Gabor Magyar • Gabor Knapp • Wita Wojtkowski
W. Gregory Wojtkowski • Jože Zupančič
Editors

Advances in Information Systems Development

New Methods and Practice for the Networked Society

Volume 1

Springer

Advances in Information Systems Development

New Methods and Practice for the Networked Society

Volume 1

Advances in Information Systems Development

New Methods and Practice for the Networked Society

Volume 1

Edited by

Gabor Magyar and Gabor Knapp
Budapest University of Technology
Budapest, Hungary

Wita Wojtkowski and W. Gregory Wojtkowski
Boise State University
Boise, Idaho USA

Jože Zupančič
University of Maribor
Kranj, Slovenia

Springer

Gabor Magyar
Budapest University of Technology
and Economics
Budapest 1111
Muegyetem rkp. 1-3.
Hungary
magyar@mail.bme.hu

Gabor Knapp
Budapest University of Technology
and Economics
Budapest 1111
Muegyetem rkp. 1-3.
Hungary
knapp@nava.hu

Wita Wojtkowski
Boise State University
1910 University Drive
Boise, Idaho 83725
USA
wwojtkow@boisestate.edu

W. Gregory Wojtkowski
Boise State University
1910 University Drive
Boise, Idaho 83725
USA
gwojtkow@boisestate.edu

Jože Zupančič
University of Maribor
Systems Development Laboratory
SI-6400 Presernova 11
Slovenia
joze.zupancic@fov.uni-mb.si

Proceedings of the 15th International Conference on Information Systems Development—New Methods and Practice for the Networked Society (ISD 2006), held in Budapest, Hungary, August 31–September 2, 2006.

Volume (1): Part 1 of a two-volume set.

ISBN-13 978-0-387-70760-0 e-ISBN-13 978-0-387-70761-7

Library of Congress Control Number: 2007929540

Printed on acid-free paper.

9 8 7 6 5 4 3 2 1

springer.com

Preface

This book is the outcome of the Fifteenth International Conference on Information Systems Development, ISD'2006, held in Budapest, Hungary between 31st August – 2nd September 2006. The theme of the 2006 conference was "New Methods and Practice for the Networked Society".

This theme expresses that we are living in a new era when practically all of our information resources are organized and managed in a networked environment. Information technology has reformed and restructured the workflows of companies and other organizations over the past several decades, and will continue to do so well into the future. This is particularly important now, as we see the emergence of complex networked information systems. "Being digital" by Nicholas Negroponte was the watchword at the dawn of the information society. "Being online" is now at the very heart of our everyday life. New postulates and requirements are stemming from this nature of society of today and tomorrow. The convergence of IT and infocommunication technologies has presented a challange for the ISD profession in terms of accomodating mobility, interoperability, the "always connected" state of information systems, the evolving distributed nature of information resources and the growing volume and diversity of information. IS development, both as a professional and academic discipline, has responded to this challenge through methodologies, tools and theory development. Progress in ISD comes from research as well as from practice. The aim of the Conference was to provide an international forum for the exchange of ideas and experiences between academia and industry, and to stimulate exploration of new solutions.

The ISD Conference evolved from the first Polish-Scandinavian Seminar on Current Trends in Information Systems Development Methodologies, held in Poland in 1988. It was a great honour and responsibility for us to organize the fifteenth event within this fine series of conferences.

Putting together a book of this magnitude requires the cooperation and assistance of many professionals with much expertise. We would like to express our

gratitude to all the authors and participants for contributing to the conference that we believe to have been successful and memorable. The conference call for papers attracted a great number of very high quality papers. All papers were double-blind refereed by at least two independent reviewers and an Associate Editor. They provided detailed reviews on all papers submitted. We would like to thank the IPC members for their essential work.

Many thanks are due also to the assistance in organization of ISD 2006, especially to the Scientific Association for Infocommunications (HTE) and to the Conference Secretary, Mr. Sándor Szaszkó. We are also grateful to the National Office for Research and Technology (NKTH) for the financial support of the Conference.

Gabor Magyar,
Gabor Knapp
Conference co-Chairs,
ISD 2006

International Science Committee

Juhani Iivari	University of Oulu	Finland
Sergey Ivanov	George Washington University	USA
Mirjana Ivanovic	University of Novi Sad	Serbia and Montenegro
Lech J. Janczewski	University of Auckland	New Zeland
Murray Jennex	San Diego State University	USA
Roland Kaschek	Massey University	New Zeland
Marite Kirikova	Riga Technical University	Latvia
Gabor Knapp	Budapest University of Technology and Economics	Hungary
John Krogstie	Norwegian University of Science and Technology	Norway
Marian Kuras	Cracow Academy of Economics	Poland
Rein Kuusik	Tallinn Technical University	Estonia
Sergei Kuznetsow	Institute for System Programming of Academy of Science	Russia
Michael Lang	National University of Ireland	Ireland
Mikael Lind	University of Boras	Sweden
Henry Linger	Monash University	Australia
Bjorn Lundell	University of Skoevde	Sweden
Audrone Lupeikiene	Institute of Mathematic and Informatics	Lithuania
Leszek A. Maciaszek	Macquarie University	Australia
Gabor Magyar	Budapest University of Technology and Economics	Hungary
Majed Al-Mashari	King Saud University	Saudi Arabia
Heinrich C. Mayr	University of Klagenfurt	Austria
Elisabeth Metais	CNAM University	France
Robert Moreton	University of Wolverhampton	UK
Pavol Navrat	Slovak University of Technology	Slovakia
Lina Nemuraite	Kaunas University of Technology	Lithuania
Anders G. Nilsson	Karlstad University	Sweden
Ovidiu Noran	Griffith University	Australia
Jacob Norbjerg	Copenhagen Business School	Denmark
Jari Palomaki	Technical University of Tampere	Finland
Malgorzata Pankowska	University of Economic Katowice	Poland
George A. Papadopoulos	University of Cyprus	Cyprus
Anne Persson	University of Skövde	Sweden
Alain Pirotte	University of Louvain	Belgium
Jaroslav Pokorny	Charles University in Prague	Czech Republic
Stephen Probert	University of London	UK
Boris Rachev	Technical University of Varna	Bulgaria

Karel Richta	Czech Technical University	Czech Republic
Kamel Rouibach	Kuwait University	Kuwait
Timothy K. Shih	Tamkang University	Taiwan
Klaas Sikkel	Universiteit Twente	Netherlands
Guttorm Sindre	Norwegian University of Science and Technology	Norway
William Song	University of Durham	UK
Ioannis Stamelos	Aristotle University	Greece
Larry Stapleton	Waterford Institute of Technology	Ireland
Eberhard Stickel	Bonn University of Applied Sciences	Germany
Uldis Suskovskis	Riga Technical University	Latvia
Istvan Szakadat	Budapest University of Technology and Economics	Hungary
Sandor Szaszko	Budapest University of Technology and Economics	Hungary
Janis Tenteris	Riga Technical University	Latvia
John Traxler	University of Wolverhampton	UK
Jacek Unold	Wroclaw University of Economics	Poland
Olegas Vasilecas	Vilnius Gediminas Technical University	Lithuania
W. Gregory Wojtkowski	Boise State University	USA
Wita Wojtkowski	Boise State University	USA
Carson C. Woo	University of British Columbia	Canada
Stanislaw Wrycza	University of Gdansk	Poland
Ilze Zigurs	University of Nebraska at Omaha	USA
Joze Zupancic	University of Maribor	Slovenia
Jozef Zurada	University of Louisville	USA

Contents

A Revised Perspective on Documentation Practices in the Modern Organisation

J. Coady, R. Pooley

Heriot-Watt University, School of MACS, Riccarton, Edinburgh.

Abstract: There are a number of reasons for the use of various methodologies in the development of systems (Broady, Walters & Hartley (1994)), notably a reduction in user dissatisfaction and more effective communication between systems developers and users. These reduce the risk of a new system being presented to its users as a fait accompli. The use of an appropriate modelling paradigm can produce a better end product, improved consistency and the likelihood of improved user acceptance.

Formal methodologies may be appropriate for more technical systems where fewer human factors are involved; however, they may be too mechanistic to be effective in detailed, day-to-day organization of developers' activities. Traditional methodologies can still be valuable in IS development projects in order to maintain an image of control or to provide symbolic status (Nandhakumar & Avison (1999)).

Hard methods, such as structured and object-oriented approaches, (Bocij et al. (1999)) focus on the Computer Based Information Systems as a technical artifact, which must satisfy a set of well-defined user requirements. According to Chekland & Howell (1998), organisational change and improvement can only be successful when the organisational actors are engaged in that change. The IS is increasingly being viewed as a social artifact, and researchers such as Stapleton (2001) and Dewar et al. (2003) highlight the need for a revised perspective on ISD to deal with this development. Coady (2003) showed that, while there is a body of work which is concerned with the social aspects of creating a technical artifact, very little work has investigated documenting the IS as a social artifact.

This paper presents an empirical study; which identifies those practices that academia suggests are the current industrial standards, compares the perceived standards to current practice in the organisations involved in the research and develops conclusions suggesting better industrial practice.

1 Introduction

Early computerized information systems were typically implemented without the use of an explicit development methodology (Broady, Walters & Hartley (1994)). As a result of early era problems developers of the 60's and 70's learnt a number of lessons, which included the use of a life cycle, consisting of documentation, control and training, and that failings could be attributed to the narrowness of perspective of the analyst and the need for a real view of the organization. These lessons led to the development of many of the current methodologies (Broady, Walters & Hartley (1994)).

A system is an organized integrated unit that serves a common purpose, formed from diverse components (Ossenbrugen (1994)). There are many modelling notations available to analysts trying to simplify systems. Systems theory generally attempts to understand the nature of systems, which are large and complex. However, systems theory suggests that whatever methodology is adopted the analyst needs to look beyond the obvious boundaries and at the system as a whole. Information systems generically have human and computer elements and both aspects are inter-related. The technical aspects are closed and predictable, whereas the human aspects are often open and non-deterministic. The technological aspects are less complex than the human aspects in information systems, because the former are predictable in nature. Many information system methodologies only stress the technological aspects. This may lead to a solution, which does not work effectively, because often methodologies underestimate the importance and complexity of the human elements (Avison & Fitzgerald (1996)). This research study examines the problems of documenting systems and reviews the frameworks used within the firms studied for best practice in the light of what is known about IS documentation.

2 The need for a revised perspective of IS documentation

The concept of the learning organisation, as presented by Argyris & Schon (1996), is defined as a means to reflect upon, and re-evaluate the knowledge that is created by individuals within the organisational context. The organisation is changed as a result of this learning process. The learning process can be viewed as an ongoing sense making activity based on the collective knowledge of the individuals (Argyris (1976)). In this way the documentation of an organisation can help in the learning and development process and create a learning organisation by continuous re-evaluation and gathering of information.

According to Chekland & Howell (1998), organisational change and improvement can only be successful when the organisational actors are engaged in that change. By implication the area of IS documentation can also be affected by this change, whereby it is a necessity that the actors in a system are involved in the documentation and modelling process in order to ensure a compatible system. 90% of Information and Communication Technology (ICT) projects fail to meet their goals due to a misalignment of goals and organisational activities (Clegg et al. (1996)). IS documentation and development is supposed to consider organisational issues but too often IS is looked upon as a subsystem external or separate from the rest of the organisation. Bednar (1999) suggests a way of improving the disparate view of IS documentation and development is to view the organisation itself as an information system. By taking this view, IS becomes an inherent part of an organisation, including its actors, and its supporting processes, and not a separate entity, thus making it easier to see the relationships and making the documentation process easier and more standardised.

The computer industry is now over forty years old. During that time it has undergone enormous change, experiencing at least three completely new generations of hardware technology (Davis (1993)). Each new generation of computing has lead to wider and wider use of computer-based information systems (CBIS). CBIS are used to address an ever-increasing variety of problems in business, as was studied in this research. As the systems have become more and more sophisticated, researchers have recognised that the modern CBIS is something more than a passive, technical artifact (Stapleton (2001)). The evidence shows that information systems are social artifacts and an integral part of the organisation (Hirscheim & Newman (1989)).

Systems development theory has however maintained a focus upon the IS as a technical artifact, and the documentation reflects this. Many of the current best practice methodologies in the IS field do not support the social aspects of the IS, i.e. the actors and entities involved, rather seeing the IS as simply a technological solution. Development methodologies and their use are at the heart of the IS development process, yet to date, very little is known about how methodologies are used within the organisational context (Fitzgerald, Russo & Stolterman (2002)). These approaches develop a software machine, which satisfies a formalised set of information-use requirements.

This implies the need for a thorough review of the traditional documentation methodologies. As we progress in the technological aspects of IS, so too must we consider the ever-changing socio-technical aspects and document and develop systems to reflect these changes. Despite a long history and considerable research into methodologies and functional specifications for information systems documentation, researchers still continue to highlight the lack of knowledge concerning the development and hence documentation of information systems. Wynkoop & Russo (1995) claim that little is known about ISD and the use of the development methodologies. They state that there has been little evaluation of methodologies in use, or examination of the selection, development, adaptation or use of methodologies in practice. It is this lack of knowledge, which has important ramifications for research and practice (Stapleton (2001)).

3 The Research Study

The task of selecting a research method, which can appropriately address and test the research issues, is of vital importance. In many ways the research instrument is a microcosm of the thesis itself, implicitly or explicitly containing the key tenets of the study (Stapleton (2001)). Given the predominance of hard systems methods over soft methods in current ISD practice and university curriculum, a study was conducted which seeks to find out:

- Are responsibilities appropriately designated within IS documentation?
- Is IS documentation developed so as to address all areas in which it is used?
- Do people use diagrams extensively in IS documentation?

The starting point of a researcher's methodological choice within information systems is not so much a problem of how many methods to employ, or if those are of a quantitative or a qualitative nature, but the ability to identify the philosophical and theoretical assumptions which leads to the choice of the appropriate methodology (Garcia & Quek (1997)). Historically, information systems research inherited the natural sciences paradigm (Mumford 1991), and as a consequence, there are a number of underlying problems associated with information systems research (Cooper (1988)). The major criticism in the past was that research tends to be dominated by "scientific approaches" (Nissen (1985), Mumford (1991)). The information system is a social artifact (Stapleton (2001) and hence the identification of the requirements for the IS involves the IS community and user groups in a difficult relationship where the world must be subjectively interpreted and reinterpreted. This perspective sees the information systems design and development process as constituting discovery and rediscovery rather than logical analysis and design (Stapleton (2001)). The research methodology incorporated a combination of quantitative and qualitative data gathering techniques, as used elsewhere in similar information systems related studies (e.g. Stapleton (2001)). A Face-to-face interview with a semi-structured questionnaire was implemented, this also allowed for any inconsistencies, uncertainties to be clarified at will.

4 Findings

A total of 19 individuals were interviewed from the IT section of five organisations. The research was concerned with studying who documents, using both standardised and non-standardised approaches, within industries and if the documentation represents all areas in which it is used. It was also concerned with addressing the issue of diagrammatic notations within documentation, why documentation is used, if it is really useful, and hence it was important to gain access to the various levels of staff within the IT sector of the company, ranging from Analysts and Documentation Engineers, to Programmers and Management.

Traditional representation paradigms assume that IT-related notations are understood and a criterion of documentation is that all areas are modeled or represented in some way in the documentation. This study aims to get an overview of the documentation project at various levels within a number of different organisations. The research therefore focuses in detail on 5 organisations, with between 3 and 6 interviewees selected in each organisation. The virtual workplace was also a valued consideration for documentation in the focus of this research; an organisation was needed, which was working virtually. Virtual environments also need to be documented in order for people to be able to co-operate effectively. This study aims to compare the findings of the one virtual organisation and the 4 non-virtual organisations in its attempt to put forward a framework for best practice. The fieldwork was carried out over a period of 2 months. The timings were suited to the organisations needs and considerations were given to timing constraints and interview deadlines adhered to. The researcher approached five organisations based on the issues uncovered in the literary review, these five organisations being well known in their sector and having to adhere to stringent standards.

As 19 individuals were interviewed regarding an IT project they had recently undertaken, the range of projects and responses varied. These projects could be summarised into 3 major categories: firstly Type 1, the Automation of Information Streaming project, secondly, Type 2, an Advanced Supply Chain Integration System Implementation and thirdly, Type 3, the Augmentation of a Supply Chain Integration System

Table 1. Categories Breakdown of IS Project by Company

Company	A				B					C				D			E		
Person	1	2	3	4	1	2	3	4	5	1	2	3	4	1	2	3	1	2	3
Type 1				X	X			X	X	X		X		X	X		X	X	X
Type 2						X	X									X			
Type 3	X	X	X								X		X						

The Table below, Table 2, illustrates the comparison of each section to the ISD literature and shows whether the findings agree or disagree with the ISD literatures view. From the Table it is apparent that in some areas agreement in more prominent than in some other areas. The area of designation of responsibilities tends to agree with the ISD literature, which suggests that organisations stick to the traditional norms when creating documentation and finding representations of their work. As stated previously however, this can cause problems if the IS is viewed solely a technical artifact and the socio-technical factors are ignored.

Table 2. Comparison of Findings with Traditional ISD views

Section	Expected Traditional ISD View	Findings
2.1 Are responsibilities appropriately designated within IS documentation?	Yes	Yes
2.2 Is IS documentation developed so as to address all areas in which it is used?	Yes	No
2.3 Do people use diagrams extensively in IS documentation?	Yes	No

As we can see from Table 2, Section 2.2 and 2.3 tend to disagree with the traditional ISD literature views. Section 2.2 questions the reusability of documentation; in this study it was found that organisations did not reuse documentation to its full capacity, something that is suggested as a priority in the literature. The ISD literature suggests that it is through this reusing of documentation organisations can learn and provide a better IS. The purpose of documentation, as addressed in section 2.2, varies in organisations, which then leads to the issue of standardisation, also addressed in within a sub-section in section 2.2. It is possibly this lack of standardisation, and use of ad hoc methodologies, which leads to the variety in use of documentation and the perspectives on documentation within the firms studied. However it is clear that irrespective of standards and methodologies that the agreement with traditional ISD views that the problem solving capabilities of the documentation justify the effort taken to create the documentation. It can therefore be suggested that tailor made documentation for organisations can work effectively.

From the Table it seems that Section 2.3, which deals with the issues of Modelling and Diagrams, tends to disagree with the traditional ISD views. Diagrams are typically suggested as a means to simplify and explain in more detail in documentation, however this study has shown that companies are reluctant to use diagrams within documentation and where they are use are ad hoc, and personalised lacking in formal standards.

The findings of this study conclusively show the lack of standardisation within the organisations in this study. This research provides a platform for reviewing the key tenets of the documentation within industry and provides a platform towards creating a best practice procedure for documenting information systems.

5 Conclusion

This study reviewed the individuals involved in the documentation process within the various organisations. This investigation is an exploratory study into the documentation processes of organisations.

The literature suggests Soft Systems Methodology as an alternative approach to documenting information systems, as opposed to the structured methodologies, within organisations. The findings presented recommend that this is the case in most organisations, with firms tending to have an ad hoc approach to documentation, even where there are standards present. It is these ad hoc methodologies that work best for the interviewees within the organisations studied and people feel more comfortable using familiar methods they can design and manipulate themselves. It is these soft systems methodologies which go partly towards viewing the IS as a social artifact and can thus aid in the learning capabilities of an organisation. This increased learning power enables the organisations to create a more synergised and effective IS, with complete documentation. Fischer and Röben (2002) state that the learning which occurs when writing manuals is the most intense form of learning, and can aid in the implementation of the IS. It can also aid in the perceptions of documentation within the organisations and encourage individuals to participate more in the creation and use of the IS documentation. It can therefore be determined that the documentation of a company should be seen as a process which can aid the organisation, rather than simply an end product. It is only through discovery and rediscovery can the organisation learn effectively to produce a better IS.

Ultimately research itself is an iterative and learning process, and the conclusions drawn out here are broad. This research suggests the soft systems methodology as a best practice methodology. Despite the vagueness of soft systems methodologies, this work suggests that these methodologies can be used effectively for the documentation needs of an organisation if tailored to suit the organisation. Future research must examine the need for a tailored approach to documentation versus mass structured approaches to documentation as taught in academia.

As stated previously, individuals who feel involved in the analysis and documentation process, even onto the design and implementation phase, accept these changes more readily. They feel ownership of some part of the IS, and hence have a reasoning to overcome any fears they may have. The documentation process does not need to be enforced in such a way that resentment is caused, a possible formalising of the present informal methodologies used being a suggestion to ensure clearer benefits needs to be highlighted to industry regarding documentation and more user involvement is necessary in order to create useful documentation, which is generic enough to be applied throughout the organisation as a whole. This ensures transferability and re-usability of the documentation within the organisation. Green & Roseman (2002) suggest that the role of the person involved in the modelling process needs to be considered more deeply when evaluating the ontological completeness requirements of the system.

This research proposes a need to revise the traditional perception of IS documentation. This project is an attempt to see documentation in a different way and introduce a better perspective of IS documentation. New perspectives imply change, a process that is not always welcomed. This change needs to include the employees and not seen to have been directed from management levels above the individuals affected by this change. This is not only true in industry but also in academia, where programmes of learning need an updating.

References

Avison, D. & Fitzgerald, G. (1988). *Information Systems Development, Methodologies, Techniques and Tools,* London: Blackwell Scientific Publications.

Argyris, C. & Schon, D.A. (1996). *Organisational Learning Π – Theory, Method and Practice*. Reading Mass: Addison Wesley.

Argyris, C. & Schon, D.A. (1976). *Organisational Learning.* Reading Mass: Addison Wesley.

Bednar, P. (1999). *Informatics – a working chaos for individuals and organisations. The impact of the notion of IS for System Analysis and Development (in Swedish)*. Lund: Dept. of Information and Computer Science, Lund University.

Bocij, P., Chaffey, D., Greasley, A. & Hickie, S. (1999). *Business Information Systems: Technology, Development & Management*, Financial Times Pitman Publishing: London.

Broady, J., Walters, S. and Hartley, R. (1994). "A Review of Information Systems Development Methodologies (ISDMS)." *Library Management* 15(6), 5-19.

Checkland, P. & Holwell, S. (1998). *Information, Systems and Information Systems: Making Sense of The Field*, Wiley, Chichester, UK.

Clegg, C. et al. (1996). *The Performance on Information Technology and the Role of Human and Organizational Factors*, Report to the Economic and Social Research Council, UK, Systems Concepts Ltd., Vol. 2 (3). Accessed April 29, 2003 at http://www.system-concepts.com/stds/clegg.html

Coady, J. (2003). Information Systems Documentation: An Empirical Study of Current Practice in Irish Firms, Thesis for Degree of Master of Science of Waterford Institute of Technology.

Cooper, R. (1988). 'Review of management information systems research: a management support emphasis', *Information Processing and Management*, 24 (1).

Davis, A. (1993). *Software Requirements: Objects, Functions and States*, Prentice Hall: New Jersey.

Dewar, RG., Pooley, RJ., Lloyd, AD., Ure, J. & Cranmore, A. (2003). Enabling knowledge sharing in collaborative design using a socio-technical pattern language, 10th ISPE International Conference on Concurrent Engineering: Research and Applications, Madeira Island, Portugal, July 2003.

Fitzgerald, B., Russo, N. & Stolterman, E. (2002). *Information Systems Development: Methods in Action*. London: McGraw-Hill Education.

Fischer, M. & Röben, P. (2002). Organisational learning and knowledge sharing: The use, documentation and dissemination of work process knowledge, *Paper for The European Conference on Educational Research (ECER) for the European Educational Research Association (EERA)* ,LISBON, 11-14th September 2002.

Garcia, L. & Quek, F. (1997). 'Qualitative Research in Information systems: Time to be Subjective?', *Proceedings from the IFIP WG8.2 Working Conference on 'Information Systems & Qualitative Research'*, Philadelphia, USA.

Green, P. & Rosemann, M. (2002). 'Perceived Ontological Weakness of Process Modeling Techniques: Further Evidence', *Paper for The European Conference of Information Systems (ECIS),* Poland, June 6th – 8th 2002.

Hirscheim, R.A. & Newman, M. (1991). 'Symbolism and Information Systems Development: Myth, Metaphor and Magic', *Information Systems Research*, 2/1, pp. 29-62.

Mumford, E. (1991). Information Systems Research-Leaking Craft or Visionary Vehicle? In *Information Systems Research: Contemporary Approaches and Emergent Traditions,* H.E. Nissen, H.K Klein, R. Hirschheim (Eds) North-Holland, Amsterdam.

Nandhakumar J. and Avison D. (1999). The fiction of methodological development: A field study of information systems development, *Information Technology and People*, 12, (2), 176-91.

Nissen, H. (1985). Acquiring knowledge of information systems - Research in a methodological quagmire. In E. Mumford, R. Hirschheim, G. Fitzgerald and A.T. Wood-Harper (Eds) *Research Methods in Information Systems*. North-Holland, Amsterdam.

Ossenbrugen, P.J. (1994). *Fundamental Principles of Systems Analysis and Decision Making,* 1-3, Wiley & Sons Inc., USA.

Stapleton, L. (2001). Information Systems Development: An Empirical Study in Irish Manufacturing Companies, Thesis for Degree of Doctor of Philosophy of National University of Ireland.

Wynkoop, J. & Russo, N. (1995). 'Systems Development Methodologies: Unanswered Questions', *Journal of Information Technology*, 10, pp. 65-73.

Towards a Dialectic Understanding of Enterprise Systems – Vendor Challenges and Contradictory Rhetoric

Stig Nordheim

Agder University College, P.O Box 422, N-4604 Kristiansand, Norway
Stig.Nordheim@hia.no

Abstract: Substantial Enterprise Systems (ES) research has focused on customers' implementation processes. This paper argues the need for a larger context to understand ES implementation. This context includes the fundamental challenge ES vendors are facing, how to satisfy unique needs with generic software. The designed solution to this challenge is analyzed, focusing on the two simple and yet fundamental concepts of commonality and variability. It is argued that the balance between commonality and variability may be viewed as a dialectic of design.

A study of ES vendors indicates that this dialectic of design is challenging. Finding an optimal balance between commonality and variability becomes an important design goal. This design challenge affects the customer directly. In the cases where variability is unable to meet the customer's requirements, a considerable pressure is applied by the vendor or implementation partner to make the customer adapt to the system. The study also demonstrates that leading ES vendors simultaneously promote both commonality and variability in their customer rhetoric, which may be viewed as dialectic. This rhetoric may even be considered paradoxical.

For customers the dialectic perspective on ES may shed light on the motivation behind the pressure applied by the vendor, and may prepare the customer for the contradictions likely to occur in the project and shakedown phases of ES implementations. For customers it is also important to understand the dialectic rhetoric in ES marketing, to be able to see through the dialectic rhetoric during the chartering phase.

1 Introduction

Enterprise Systems (ES) are commercial software packages that enable the integration of transaction-oriented data and business processes throughout an organization (Markus and Tanis, 2000). Two types of ES are studied here, Enterprise Resource Planning (ERP) and Enterprise Content Management (ECM).

ES represent an effort to apply generic software to meet the requirements of heterogeneous organizations. They may be viewed as a response to the challenge aptly formulated by Markus (1997): “the world around us is demanding that we find ways to satisfy unique needs with generic software and components”. One may argue that the ES effort to satisfy unique needs with generic software implies a potential contradiction. This is a reasonable assumption as several studies have established the contradictory nature of ES implementations (Besson and Rowe, 2001; Robey et al., 2002; Soh et al., 2003; Wei et al., 2005).

A set of implementation choices may be viewed as a synthesis of the contradictory nature of ES implementations (Nordheim and Päivärinta 2006). Some generic ERP implementation choices are summarized by Luo and Strong (2004). These include three technical ERP customization options: module selection, table configuration and code modification.

This paper views the contradictory nature of ES implementations in a larger context, i.e. how ES vendors approach the challenge of solving the requirements of unique organizations with generic software solutions. The reason for focusing on such a larger context for ES implementations is two-fold. First of all, ES implementation involves a considerable knowledge transfer effort from the vendor to the customer (Lee and Lee, 2000). Secondly, according to Markus and Tanis (2000) misalignment issues and their resolutions in one phase may originate in previous project phases. An understanding of the vendor challenges may therefore be relevant for customers as a larger context for ES implementation.

There are two simple and yet fundamental design concepts used by ES vendors to address the challenge formulated by Markus (1997), namely the distinction between commonality and variability (C/V) (Leishman, 1999). Due to the contradictory nature of ES implementations, this paper takes a contradictory view of C/V in ES design, and raises the following question:

In a dialectic perspective, what are the ES vendor challenges related to the commonality/variability design issue, and what are the possible implications for ES customers?

Considerable ES research has focused on customer implementation issues, including dialectics (Besson and Rowe, 2001; Robey et al., 2002; Soh et al., 2003; Wei et al., 2005). Few studies have focused on ES vendors, one exception is Liang and Xue (2004).

First the concepts of commonality and variability (C/V) are presented, followed by the concepts of dialectics. An argument for a contradictory view of C/V is then presented. The empirical part is an interpretive study of nine ES vendors. The implications are discussed from an ES customer perspective, in relation to the ES implementation process and knowledge transfer.

1.1 Commonality and Variability in Enterprise Systems

Commonality and variability (C/V) is a characteristic of most software packages (Bühne et al., 2005). The general meaning of commonality is properties shared by all members of a group, and in software engineering commonality is expressed as "an assumption held uniformly across a given set of objects" (Coplien et al., 1998). Commonality is designed as properties shared by all customers of an ES.

Variability is generally understood as properties varying within members of a group, and may be viewed as "an assumption true of only some elements of a given set of objects" (Coplien et al., 1998). In a software engineering context, variabilities are "bound" by placing specific limits on each of the variabilities. One example of bounded variability may be a range of legal values for a parameter (Coplien et al., 1998). Variable properties are designed to be changed by different customers of an ES.

Within ES the designed commonality may include: common business processes, functions, workflows, screens, technical infrastructure layer, data models, and common default parameter settings (Leishman, 1999). As an example, SAP's variability includes the organization model, the process model, the function model, the data model and subsequent table settings, how applications and services are distributed across the computational tiers, and the user interface of screens and screen flows. It also includes interoperability with other programs, and the ABAP/4 programming environment (Leishman, 1999).

ES represent both bounded variability (e.g. configuration) and variability that is not bounded, (e.g. user exits and programming). For a customer it is interesting to note that bounded variability is associated with only slight maintenance efforts, whereas variability that is not bounded is associated with heavy maintenance efforts (Brehm et al., 2001). Customization is one example of variability that is not bounded, and to avoid code customization is a strategic critical success factor (Somers and Nelson, 2001).

Different vendors will, of course, reach very different decisions about the right balance between C/V (Davenport, 1998). This is also shown in the cases presented later.

1.2 Dialectics

Dialectics is a way of thinking, based on contradictions. A contradiction can be seen as a relation between two opposite aspects of a phenomenon, called thesis and antithesis; where antithesis is the negation of the thesis. The two aspects of a contradiction are intrinsically related, yet opposite and distinct from one another, and one aspect in a contradiction cannot be fully understood without considering the other (Van de Ven and Poole, 1995).

In dialectical theory, stability and change are explained by reference to the balance of power between the two opposing entities. A thesis (A) may be challenged by an antithesis (Not-A), and the resolution of the conflict becomes a synthesis (which is Not Not-A). By its very nature, the synthesis is a novel construction that departs from both the thesis and the antithesis. Dialectics is about dynamics, as dialectical theory is one way of explaining development and change (Van de Ven and Poole, 1995).

According to Dahlbom and Mathiassen (1993), contradictions can in some cases surface as trade-offs: "From a dialectical perspective, these trade-offs are manifestations of contradictions inherently related to the use and development of computer systems" (p63). Dialectic thinking is applied to the C/V concepts here, since dialectics contributes to the production of knowledge by an increased understanding of a phenomenon (Israel, 1979).

1.3 A dialectic view of C/V

Mechanisms for C/V are built into ES (Leishman, 1999), and commonality may be formulated as a thesis of ES design: "certain ES properties should not be subject to change". The thesis of commonality represents stability, and is true for a subset of the ES properties. The design idea behind commonality is that one should not change a certain property of the system, and one example is the ERP argument that the system contains "best practices" which should not be changed.

Variability may be formulated as an antithesis of ES design: "certain ES properties should be subject to change". The antithesis of variability represents change, and is true for a subset of the ES properties. The design idea behind variability is that one should be able to change a certain property of the system, as each customer is unique and requires a solution adapted to its requirements. Customers vary and their contexts may be different, therefore the system needs to have variable properties.

So for each ES property there is a design decision: whether the property should be subject to change or not. This design decision can therefore be viewed as contradictory, where one decision represents the opposite of the other (Figure 1).

"Property x should not be subject to change" (The thesis of Commonality)

$\updownarrow$

"Property x should be subject to change" (The antithesis of Variability)

Fig. 1. The C/V design decision viewed in a dialectic perspective

If one accepts the thesis and antithesis as expressed in Figure 1, the C/V design decision of ES may be viewed in a dialectic way. An ES may then be viewed as a synthesis: some properties of the system can and should be subject to change, other properties of the system should not be subject to change. A design with 100% commonality would imply a rigid system, easy to install, but with an inflexibility that is completely unacceptable for a complex ES. A design with 100% variability would imply a total flexibility where every property had to be specified, a formidable installation effort and a completely meaningless situation for a complex ES. The synthesis is therefore a balance between commonality and variability, and represents the vendor's effort to satisfy unique customer needs with generic software. This study explores how the C/V balance is viewed by ES vendors, and possible consequences for ES customers.

2 Method

The study has been guided by the principle of the hermeneutic circle, combined with the principle of dialogical reasoning (Klein and Myers, 1999). Following an initial literature review, interviews were made with representatives from two ES vendors and one implementation partner. The two ES vendors were eZ Systems, one larger ECM vendor; and MultiPlus, a smaller ERP vendor. The implementation partner represented Agresso, a larger ERP vendor.

Six informants were interviewed, two for each system. They had 5-15 years' implementation experience. The interviews have been carried out by the author as qualitative, open interviews based on an interview guide. Dialectics was used as a "sensitizing concept" (Patton, 1990) to guide data collection and analysis. The interviews lasted ca 40 minutes each; they were audio-taped and transcribed. Using Atlas.ti for data analysis, the interviews were coded according to the following categories: commonality, variability, the C/V balance, adaptation of system and adaptation of organization.

The findings were followed up with a document analysis of vendor statements published on the web. Six dominant vendors were selected, to supplement the interview data. These included SAP, together with five major ECM vendors presented in the CMS report (CMS Watch, 2006) as providing enterprise platforms. Table 1 provides an overview of the data sources used.

Table 1. Data sources used in the study

Type of ES	Vendor	Type of data
ERP	Agresso	Interviews, transcribed and coded
	MultiPlus	Interviews, transcribed and coded
	SAP	Web documents
ECM	eZ Systems	Interviews, transcribed and coded
	FileNet	Web documents
	Vignette	Web documents
	Documentum	Web documents
	Interwoven	Web documents
	Stellent	Web documents

3. Cases

Two of the three cases where interviews were carried out, are described, since they provided interesting insights. The third case was a small ERP vendor, MultiPlus (www.multiplus.as). This case confirmed the findings from the two first cases, without adding new insights. Hence, it is left out of the description. The six vendors which were subject to document analysis are not described, only web references are given for these.

3.1 Agresso

Case number one is the Agresso ERP system (www.agresso.com), based in the Netherlands. With more than 2300 customers in 70 countries, Agresso offers solutions both to public and private sectors. Two interviews were carried out with the Agresso product manager at one of their main implementation partners in Norway. Another interview was with a hired consultant who was the project manager for a large Agresso implementation project. He had 15 years of experience with implementing Agresso for customers.

Commonality is heavily emphasized by Agresso. This is seen in their emphasis on complete solutions for public and private sectors. Variability is also emphasized, and is mainly provided by frameworks. Frameworks include customization and configuration tools. According to the informants, frameworks are only customized at national and business sector levels. At the customer level frameworks are in principle not customized, and the only variability mechanism recommended at customer level is configuration. Variability is also described as a "templated approach" to implementation (Agresso, 2005).

The Agresso product manager pointed out what he considered a puzzling phenomenon. On the one hand, Agresso provides a powerful customization tool that includes VBA interfaces (Visual Basic for Applications). On the other hand, Agresso emphasizes the importance of a limited configuration. According to the informant, this may indicate a rather ambiguous attitude to variability.

According to the informants, it appears that the more flexible the ES is in terms of configuration, the more customer pressure there is likely to be to configure rather than change the organization. Handling such pressure is considered a vital role of the consultant. From an implementation partner perspective, variability should not automatically imply an adaptation of the ES. The Agresso product manager advised against a mere ES configuration without a preceding organizational development process. As an illustration the Agresso implementation at the municipality of Oslo was cited, where the need for reports was reduced from around 1600 to 100.

3.2 eZ Systems

Case two is eZ Systems (http://ez.no/), an Open Source ECM vendor with 30-40 thousand downloads per month. eZ Systems' idea is to provide flexible solutions rather than "off-the-shelf" software. One of the informants has a split role between management of customer projects and programming, the other informant is responsible for all customer projects in the company.

Commonality is a kernel that is developed and controlled exclusively by eZ Systems. According to the informants, an implementation based on as much commonality as possible means better maintainability, higher quality and a cheaper solution.

There are two types of variability provided by eZ Systems' development framework; referred to as "supported" and "unsupported" variability by the informants. Supported variability consists of modifying HTML templates and configuration. Unsupported variability consists of interface programming, usually based on existing libraries. This variability is achieved by plug-ins into the kernel, and as the system evolves, plug-ins are extended without affecting the kernel.

Commenting on the C/V balance, the vendor's goal is to have as much configurability and as little programming as possible. The ideal is to empower non-programmers to establish complex solutions by simple configuration. A goal is to cover 95% of a customer's needs by configuration. Configuration constitutes the typical implementation effort, together with modifying HTML templates. There is hardly any programming in a typical implementation, although larger implementations are characterized by some programming.

The demarcation line between C/V is perceived by eZ Systems as an interesting design trade-off. As pointed out by one of the informants, if there is too much commonality compared to variability, the customer is being locked up due to lack of configuration options. But if there is too much variability compared to commonality, the customer will be confused due to the lack of standard functionality.

Consultancy on variability is part of eZ Systems' business model, and ranges from adapting templates for small businesses, to complex integration with legacy systems in large enterprises. Consultancy attempts to influence the customer, so that requirements can preferably be met by configurable variability.

3.3 SAP

This ERP vendor emphasizes commonality, that they provide complete solutions: "Building on the ground-breaking idea of standard enterprise software, we have become the leading provider of complete business solutions…" (SAP, 2003). In addition to a huge portfolio of software code, some commonality is represented by the so-called best business practices. These are claimed to be based on industry knowledge gained from nearly 19,000 customers in more than 20 industries (SAP, 2005).

SAP also emphasizes variability: "To deliver real value, your solutions have to be as unique as your business. That often means company-specific functionality and modifications – which can easily lead to spiralling costs. That's why we offer a range of dedicated custom-development services… This helps you get the most out of your investments in tailor-made solutions, while enhancing your competitive edge" (SAP, 2003b). One of the services of their more than 9000 SAP consultants is custom development (SAP, 2003b). There is also one example of reduced variability, the "mySAP All-in-One" solution, built to fit small customers. This is a pre-configured, industry-specific version (www.sap.com/solutions/sme/). SAP argues for its commonality by promoting best practices, and argues for its variability by promoting the services of its 9000 consultants to achieve unique customer solutions. SAP also promotes commonality and variability simultaneously*:* "Through extendability our customers can gain competitive advantage and have access to a cost-efficient mix of standard functionality and custom development" (SAP, 2003).

3.4 Five ECM Vendors

All the five ECM vendors studied (cf. Table 1) emphasize commonality, that they provide complete solutions. Commonality is also said to have the benefit of transforming the organization: “The true business value of Enterprise Content Management (ECM) emerges when it transforms an organization’s operations to best meet the specific needs of its industry” (FileNet, 2006). Commonality is sometimes referred to as “out-of-the-box” software that allows customers to quickly create, update, manage and deploy virtually any type of electronic asset (Vignette, 2006).

All the five ECM vendors also emphasize the importance of variability, highlighting their adaptable and flexible software. Configuration, integration and consulting services are aspects of variability that are typically emphasized, and also custom code: “As with all business applications, certain business requirements may be entirely unique to a particular company or system environment. As a result, it often becomes necessary to construct custom code modules” (FileNet, 2006b). “ECM Documentum Consulting provides customers with highly customized solutions that support their unique combinations of platforms and applications… When business, process, or platform specifications dictate the extension of product or integration functionality beyond configurable capabilities, ECM Documentum consultants can apply programming expertise to … tailor the platform to the specific business case or technical environment.” (Documentum, 2006).

4 Discussion

According to Markus and Tanis’ (2000) enterprise system experience life cycle, the origins of misalignment issues and their resolutions may be found in previous project phases. Following the principle of contextualization (Klein and Myers, 1999), it is reasonable to consider an even larger context of the ES experience; i.e. the origins of misalignment issues in ES implementations and their resolutions may be due to vendor challenges. This paper considers two simple concepts in this larger context, the C/V balance. This is at the heart of the ES attempt to satisfy unique needs with generic software. The C/V design decision (Figure 1) highlights an important design challenge for ES vendors, and provides concepts for discussing important issues at the heart of ES design. The main findings are summarized in Table 2.

Table 2. Summary of findings: ES vendor challenges related to C/V

Issue	Description
The C/V design challenge	ES vendors build C/V mechanisms and view the C/V balance a design challenge (“a dialectic of design”).
Commonality	Commonality is emphasized by all the vendors studied, and

emphasis	ES are presented as complete solutions, e.g ERP best practices.
Variability emphasis	Variability is emphasized by all the vendors studied, that ES can fit unique customer needs. To provide sufficient variability is a vendor concern, especially configurability. For the customer it is crucial that the variability is supported by the vendor. Consulting services also promote variability.
Pressure related to variability	When variability (e.g configuration) can solve a requirement, customers apply pressure on the vendor to configure rather than adapt the organization. When variability can <u>not</u> solve a requirement, vendors apply pressure on the customers to modify their requirements to a configurable solution ("a dialectic of adaptation").
Both commonality and variability are promoted	Vendors simultaneously promote both commonality and variability aspects of ES in their marketing ("a dialectic rhetoric").

The C/V design challenge (Table 2) is to find an optimal balance between stability and change. In the eZ Systems case this is perceived as an interesting design trade-off. Such design trade-offs are indeed manifestations of contradictions inherently related to the development of the systems, as pointed out by Dahlbom and Mathiassen (1993). This C/V design challenge is here labelled "a dialectic of design", since dialectics is about dynamics (Ven de Ven and Poole, 1995), and the C/V demarcation line is dynamic. Viewed as a response to Markus' (1997) statement on how to find ways to satisfy unique needs with generic software, the dynamic C/V balance may be appreciated as a positive and important driving force for the development of ES. eZ Systems' goal to have as much configurability as possible is an example of a quest for increased variability, and SAP's "mySAP All-in-One" is an example of a quest for reduced variability. As ES vendors have chosen to balance C/V in different ways, the synthesis will presumably have to be dynamic, with no final solution.

The commonality emphasis of complete solutions is characteristic of ES, but they are hardly ready to be installed "out-of-the-box" as e.g Vignette (2006) claims. The variability emphasis of ES as "unique solutions", raises a question for customers: what kind of variability is promoted? Is it vendor-supported table configuration or unsupported code modification (cf. Luo and Strong, 2004)? To provide sufficient variability by configuration is a vendor concern. When consultancy is promoted, is this to remedy a lack of variability, or is consultancy necessary to handle the complexity of ES variability mechanisms?

The pressure related to variability (Table 2) means that the C/V demarcation line is important for the customer too. This raises a question concerning the knowledge transfer effort from vendor to customer during implementation (Lee and Lee, 2000). If vendors focused on the C/V demarcation line in their communication with customers, would knowledge transfer to the customer improve?

Further research should explore whether an explicit focus on the C/V demarcation line would facilitate knowledge transfer to the ES customer.

Both commonality and variability are promoted simultaneously, in what may be labelled a kind of “dialectic rhetoric” (Table 2). This is also done in practice, as with Agresso’s dual emphasis on complete solutions and at the same time providing a powerful customization tool. Another example is SAP, promoting best practices, and yet promoting the services of more than 9000 consultants. ES consultancy has become a lucrative business that thrives on the challenge to satisfy unique needs with generic software.

The dialectic rhetoric raises a question about the dialectic view of the C/V design (Figure 1): that for each ES property there is an “either-or” decision. This contradicts the vendor rhetoric, where the sum of properties are presented as a “both-and”. Which one is true? Is C/V a dichotomy or a continuum? This is a question of perspective. Considered at a detailed level, the C/V design decision is dialectic, each individual property is either subject to change or not (Figure 1). At an aggregate level however, the total set of properties may be viewed as a continuum, due to the large number of properties. The aggregate “both-and” perspective presented by the vendors, may be problematic for customers during the chartering phase of implementation. The ES chartering phase tends to be dominated by a deterministic vision (Besson and Rowe, 2001), and understanding some vendor rhetoric as contradictory “either-or”, may enable the customer to see beyond the rhetoric. Further research is needed to establish how the dialectic rhetoric is perceived by customers, and whether it contributes to the chartering phase sense of technological determinism.

5. Conclusion

Based on two simple concepts, the C/V balance focuses on a key issue when unique needs are to be satisfied with generic ES software. To consider this larger context of the ES experience has the following implications. Viewing the underlying C/V design as a dialectic “either-or”, may enable ES customers to see through the vendors’ “both-and” rhetoric during the chartering phase. The C/V demarcation line also affects the customer directly. If variability is unable to meet the customer’s requirements, a considerable pressure is likely to be applied by the vendor or implementation partner to make the customer adapt to the system. For customers therefore, it is important to view the C/V design as dialectic and focus on the scope of variability, thus to be prepared for contradictions likely to occur in the project and shakedown phases of ES implementations.

References

Agresso (2005) Whitepaper: “Implementation the easy way”. www.agresso.com.

Besson, P. and F. Rowe (2001). "ERP project dynamics and enacted dialogue". SIGMIS Database 32(4): 47-66.
Brehm, L., Heinzl, A., Markus, M.L. (2001). "Tailoring ERP Systems: A Spectrum of Choices and their Implications". Proceedings of HICSS-34, Los Alamitos CA, IEEE.
Bühne, S., K. Lauenroth, et al. (2005). "Modelling Requirements Variability across Product Lines". 13th IEEE International Conference on Requirements Engineering, Paris, France, IEEE Computer Society.
CMS Watch (2006). http://www.cmswatch.com/CMS/Report/Vendors/
Coplien, J., Hoffman, D., and Weiss, D. (1998). "Commonality and Variability in Software Engineering". IEEE Software. 15(6): p. 37-45.
Dahlbom, B. and L. Mathiassen (1993). Computers in Context: The philosophy and practice of systems design. Cambridge, Mass., NCC Blackwell.
Davenport, T. H. (1998). "Putting the Enterprise into the Enterprise System." Harvard Business Review 76(4): 122-131.
Documentum (2006). http://www.documentum.com/consulting/custom_dev.htm
FileNet (2006). http://www.filenet.com/English/Industry_Solutions/index.asp
FileNet (2006b) Consulting Services. http://www.filenet.com/English/Customer_Center/Professional_Services/Consulting_Services/Consulting_Services/012050005.pdf
http://ez.no/
Israel, J. (1979). The Language of Dialectics and the Dialectics of Language. Copenhagen, Munksgaard.
Klein, H. K. and M. D. Myers (1999). "A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems." MIS Quarterly 23(1): 67-93.
Lee, Z. and J. Lee (2000). "An ERP implementation case study from a knowledge transfer perspective." Journal of Information Technology 15: 281-288.
Leishman, D.A. (1999). Solution Customization. IBM Systems Journal, 38,1: 76-97.
Liang, H. and Y. Xue (2004). "Coping with ERP-related contextual issues in SMEs: a vendor's perspective." Journal of Strategic Information Systems 13: 399-415.
Luo, W. and D. M. Strong (2004). "A framework for evaluating ERP implementation choices." IEEE Transactions on Engineering Management 51(3): 322-333.
Markus, M. L. (1997). "The Qualitative Difference in IS Research and Practice". Proceedings of the IFIP TC 8 8.2 International Conference on Information Systems and Qualitative Research, Philadelphia. 11-27.
Markus, M. L. and C. Tanis (2000). "The Enterprise System Experience - From Adoption to Success". R. W. Zmud (ed.): Framing the domains of IT management: projecting the future through the past. Ohio, Pinnaflex: 173-207.
Nordheim, S. and T. Päivärinta (2006). "Implementing Enterprise Content Management: From Evolution through Strategy to Contradictions Out-of-the-Box." European Journal of Information Systems. In press.

Patton, M. Q. (1990). Qualitative evaluation and research methods. Newbury Park, Sage.
Robey, D., J. W. Ross, et al. (2002). "Learning to Implement Enterprise Systems: An Exploratory Study of the Dialectics of Change." Journal of Management Information Systems 19(1): 17-46.
SAP (2003). "SAP Makes Innovation Happen". SAP AG Brochure Material No. 50 065 210.
SAP (2003b). "SAP Customer Services Network". SAP AG Brochure Material No. 50 045 557.
SAP (2005). "Integrated Industry-Specific Enterprise Solutions For Midsize Businesses". AG Brochure Material No. 50 061 060.
Soh, C., S. K. Sia, et al. (2003). "Misalignments in ERP Implementation: A Dialectic Perspective." International Journal of Human-Computer Interaction 16(1): 81-100.
Somers, T. M. and K. Nelson (2001). "The impact of critical success factors across the stages of enterprise resource planning implementations". Proceedings of HICSS-34, Los Alamitos CA, IEEE.
Van de Ven, A. H. and M. S. Poole (1995). "Explaining development and change in organizations." Academy of Management Review 20(3): 510-540.
Vignette (2006) http://www.vignette.com/contentmanagement/0,2097,1-1-1928-4149-1966-4676,00.html
Wei, H.-L. H., E. T. G. E. Wang, et al. (2005). "Understanding misalignment and cascading change of ERP implementation: a stage view of process analysis." European Journal of Information Systems 14(4): 324-334.

Understanding Strategic *ISD* Project in Practice – An *ANT* Account of Success and Failure

Rebecca Abrahall*, Dubravka Cecez-Kecmanovic*, Karlheinz Kautz**

*School of Information Systems, Technology and Management, Faculty of Commerce and Economics, UNSW, Sydney, NSW 2052, Australia
dubravka@unsw.edu.au
**Copenhagen Business School, Department of Informatics, Howitzvej 60, DK-2000 Frederiksberg, Denamrk, and School of Information Systems, Technology and Management, Faculty of Commerce and Economics, UNSW, Sydney, NSW 2052, Australia, Karl.Kautz@cbs.dk

Abstract: This paper presents an interpretive case study of a strategic information system development (ISD) project in an insurance Company whose outcomes were perceived as both a success and a failure. By following actors – both human and non-human – involved in the strategic ISD project and the processes of inscribing and aligning interests within their actor-networks, the paper aims to unpack and provide a rich description of the contradictory nature of the socio-technical in such a project and the making of its success and failure. Guided by Actor Network Theory (ANT) the description traces the emergence of heterogeneous actor-networks and reveals how and why some interests did translate while others didn't into the IS designs, thereby producing the perceptions of success or failure.
Keywords: Strategic ISD, IS success, IS failure, Actor Network Theory, ISD as heterogeneous actor-networks

1 Introduction

The value and importance of strategic information systems (IS) – defined as systems that alter a firm's processes, products and/or services and change the way a firm competes in its industry – has long been recognized by industry practitioners and academics alike [5]. Many studies focusing specifically on strategic IS success or failure recognized the importance of social factors [8], implying a greater need for richer approaches to and deeper understanding of strategic ISD processes. The prescriptive solutions offered by the predominantly functionalist, positivist perspectives fail to offer a deeper understanding of complexities and subtleties involved in the strategic ISD processes in practice, especially perceptions of their successes and failures [8], [15]. Furthermore, it has been emphasised that in order to achieve deeper understanding both the socio-political and the technical nature of strategic ISD need to be investigated in an integrative way [2], [17], [13], [15].

Although a significant body of IS literature investigates social and technical issues, the bulk of this literature simplifies the ISD project environment into two segregated domains – the social and the technical. Many argue that such views are too simplistic to account for the complex nature of both IS strategising and ISD, calling for a more holistic research approach that better accounts for the inner-workings and intricacies of these vital business processes [23], [21], [3], [6], [15], [12]. But to do that we need to address the very nature of the social and the technical as they merge in ISD projects, which has been the subject of ongoing struggles in the IS literature [23].

In this paper we investigate a strategic ISD project in an insurance Company with the aim to i) provide a rich description of the socio-technical nature of such a project, and based on this description ii) improve understanding of the socio-technical interplay between actors and explain how this interplay impacts the perceived success and failure of the project. To achieve these objectives we use Actor-Network Theory (ANT) as a theoretical lens to investigate and explain the nature of socio-technical work and the interplay between human and non-human 'actors' throughout the strategic ISD project (see e.g. [4], [9], [10]). Following a brief description of some key concepts of ANT, we present research design and the interpretivist case study of the strategic ISD. The empirical data (interviews, researcher's notes and documentation) were then analysed and interpreted through the lens of ANT thus enabling deep insights into the socio-technical nature of ISD and the resulting perceptions of both success and failure of the system.

2 Theoretical Background: Some Key Concepts of ANT

By rejecting the traditional sociological view of the 'social' as a particular domain of reality used to provide explanations of science, technology and society, ANT aims to explain the social by tracing 'associations' among heterogeneous actors as they interact and form more or less durable wholes – actor-networks [4], [9], [10]. Also called the 'sociology of association', ANT assumes no a priori distinction between human and nonhuman actors, and sees them as active makers of actor-networks. ANT offers a uniform framework that accounts for micro, meso and macro levels of analysis, without privileging any [14]. ANT has been used in IS research to "study the social relations and processes by which [an IS] is fabricated, [considering] the facts and artefacts which mediate and reinforce those relationships" ([3], p. 200). Its central concern is to understand and theorise the role of technology and technological objects in making the social [6], thus enabling deeper understanding of ISD success and failure in strategic projects [8]. Examples of ANT research include Mitev's analysis of the new ticket reservation system at the French Railways [14]; Aanestad's investigation of the impacts of tele-medicine infrastructure in surgery [1]; and Walsham and Sahay's research into the adoption of GIS for district-level administration in India [24].

ANT is an emerging body of work and makes no a priori assumptions about the social world[1]. Some core ANT concepts, which remain constant throughout the body of literature, are summarised in Table 1. ANT is based on the core concepts of the actor. An actor is an entity – human, nonhuman or a combined hybrid object of the two – that can affect action in an actor-network. An actor-network is a heterogeneous network of aligned interests working toward the achievement of a common goal. The alignment of interests within an actor-network is formed through the enrolment of a body of allies (who become actors – both human and nonhuman) through a process of translating their interests to be congruent with those of the network [23]. This translation is achieved by inscribing actors' interests in the new system using 'scripts', which influence actors to assist an actor-network in the achievement of its goals. The act of inscribing actors with the necessary scripts is referred to as a program of action. Conversely, the act of challenging these programs of action is referred to as an anti-program of action [19]. These interests are inscribed into delegates, which are actors that stand in and speak on behalf of particular viewpoints that have previously been inscribed in them [24].

Table 1. Core concepts of ANT (adopted from [23], p. 468)

ANT concepts	Description
Actor or actant	Both human beings and nonhuman actors such as technological artifacts, documents, objects, etc.
Actor-Network	Heterogeneous network of aligned interests, including people, organisations and technology
Enrolment & Translation	Creating a body of allies, human and nonhuman, through a process of translating their interests to be aligned with the actor-network
Delegates & Inscription	Delegates are actors who 'stand in for' particular viewpoints which have been inscribed in them, e.g., software as a frozen organisational discourse
Irreversibility	The degree to which it is subsequently impossible to go back to a point where alternative possibilities exist
Black-Box	A frozen network element, often with properties of irreversibility
Immutable Mobile	Network element with strong properties of irreversibility and effects which transcend time and place, e.g., software standards

There are no prescriptive recommendations for the use of ANT as a research methodology. The following except from Walsham [23] is perhaps the best illustration and justification for the rationale behind ANT as a methodology:

> [ANT] is both a theory and a methodology combined…[as] it not only provides theoretical concepts as ways of viewing elements in the

[1] ANT has been criticised on several grounds (especially the symmetrical treatment of human and nonhuman objects). These criticisms will not be addressed in this paper, however a full analysis can be obtained from [23].

> real world, it also suggests that it is exactly these elements which need to be traced in empirical work (p. 469).

Walsham and Sahay [24] noted that the aim of ANT is to examine the motivations and actions of actors in heterogeneous networks of aligned interests, by following these actors (and the work they do) through the actor-network. Underwood suggests that by following the actors of interest in a network and describing what we see as the key to revealing association s they make up the social, political, technical and contextual situations [20]. As a research methodology, we have chosen to enact the ANT methodology in this way, in order to trace actors through actor-networks, describing emergent situations using ANT terminology in order to understand, describe and ultimately reveal a rich description of a strategic ISD process and its outcomes.

3 Research Method

In enacting ANT methodology, we conducted an interpretive case study of a strategic ISD project – including a field study and a historical reconstruction of the project since its inceptions. Interpretive case study research was selected because it allows for tracing the associations and the construction of meaning through the direct engagement between the researcher and the actors – both human actors who experienced first-hand the situations being investigated and nonhuman actors involved in these situations. Secondly, achieving a rich understanding of the complex nature of a strategic ISD project necessitates that the study is conducted within its natural setting [22]. The case study was partially historically reconstructed as the key phase of the project studied was completed before the research started and some important actors left the Company.

The case selection required a strategic ISD project with an appropriate level of risk and complexity such that a degree of richness in data could be assured [16]. The selected Company that we call Olympia is the Australian arm of a large international insurance company. The project was unique in that it was an industry-first e-commerce system that transacted the Company's business insurance product direct to their brokers over the web. It was also interesting that the outcomes of the project were considered an outstanding success in the marketplace, however internally it was resented and considered a failure for not delivering required functionality and for being over-budget and over-time.

The case selected was auspicious in that one of the authors had previously worked at the Company over a six-month period as a member of the project team. The subsequent field study followed the actors: developers, managers, users as well as various technologies, plans, and documents. Empirical data gathered include i) transcripts of eleven interviews with two Architects, two Application Developers, Test Team Leader, Data Migration Developer, Senior Business Analyst, Business Expert– Underwriting, Business Project Manager, Senior Information Systems Executive, and Business Expert–Brokers; ii) project documentation (including historical documents); and iii) researcher's notes after the interviews and informal conversations with former colleagues.

The first stage of data analysis involved reading through printed copies of interview transcripts, notes and documents, highlighting interesting texts and tentatively classifying them under broad categories or 'codes' (open-coding). By following the actors – developers, analysts, managers, project plans, technology platform, strategic IS, etc., and by tracing their association and actor-networks' formation the analysis expanded, necessitating redefinition and (re)grouping of codes and sub-codes. Through an iterative process codes/sub-codes and related quotes were then arranged, on a 3x4m paper on a wall (dubbed the ANT wall) in a large office indicating various associations and actor-networks. Such a comprehensive visual representation of the findings enabled further exploration of the interplay between different human and nonhuman actors and dynamics of their associations within a bigger picture.

We approached theoretical interpretation by first identifying and making sense of key events and points in time throughout the project that for whatever reason were considered important to the actors and project outcomes. Through an iterative process of describing and examining the emergence of these events using ANT semiotics we traced the socio-technical associations, alignment of interests, inscription and translation, operating throughout the strategic ISD project.

4 The Case Company and its Industry Context

Olympia is (a pseudonym for) the Australian arm of a large multinational financial services institution, dealing primarily in general business and life insurance. In 2001, Olympia's General Insurance (GI) Business Division undertook to become the first insurance provider in Australia of web-based e-business services to their Broker Community (clients), selling their business insurance products online. Prior to the development of this Information System, named 'Olympia-Online', Olympia was not taken seriously, or seen as a major competitor in the Australian general insurance market. All e-business in the Australian Insurance Industry was conducted via 'BrokerLine', an outdated mainframe-based electronic platform, run by Telcom, an Australian telecommunications company. More so than any of their competitors, this platform was vital to Olympia since all its business is mediated through Brokers, and Olympia has no direct contact with individual customers in the general insurance domain.

In 2001 Telcom announced to the Insurance Industry that they were ceasing operation of the current e-business platform (BrokerLine) and all companies were required to move their business operations to 'Horizon', a new web-based platform. This situation is presented by an actor-network in Fig. 1 that shows the Telcom Company exercising influence on Australian insurance companies to transfer their business from BrokerLine over to Horizon and in doing so attempting alignment with the Broker community. Unlike Olympia, most insurance companies transacted their business both directly with individual business and via the brokers (indicated by their reciprocal alignment with both). This is why Olympia was particularly vulnerable to the change of the Telcom platform.

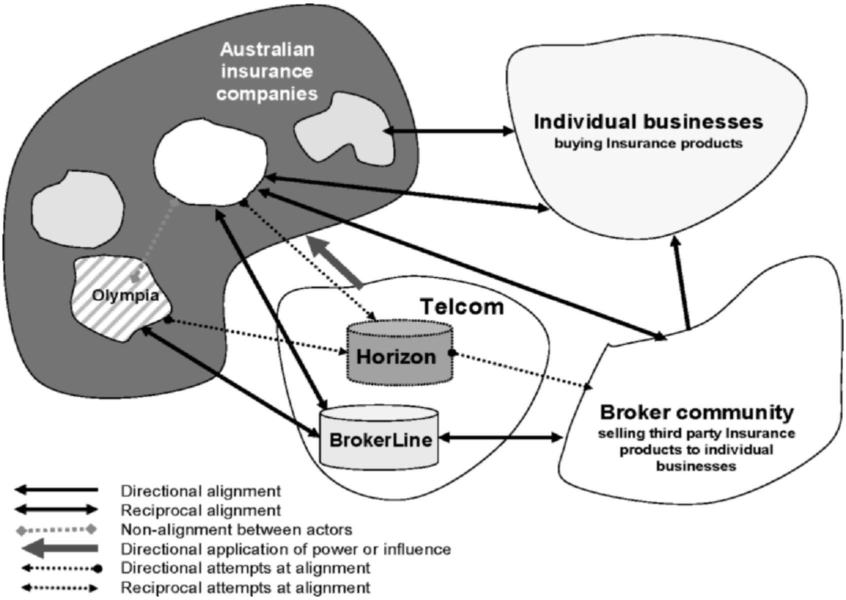


Fig. 1. An actor-network describing the situation in the Australian Insurance Industry early-mid 2001

Fearing the loss of all their business, and simultaneously recognising the opportunities a new web-based platform afforded, Olympia's GI Business Division and Strategy and Planning Division went about putting together a business case for the development of a new web-based e-business system, Olympia-Online. By inscribing Olympia's interest and its new strategy into the Business Plan for Olympia-Online development and by charging Information Services Department with the responsibility to develop a concrete solution (a new information system to interface with the Horizon's web-based platform) GI Business Division and Strategy and Planning Division succeeded to enrol Information Services Department into the new actor-network. This inscription seems to be strong enough to motivate Information Services Department to attempt alignment with Horizon. With a prospect of becoming the only channel through which Olympia would interact with the brokers in order to sell its insurance products, Olympia-Online development became a key strategic IS project in the Company.

5 The Story of Olympia-Online Development

Olympia-Online was a new type of information system in the Insurance Industry of a magnitude never experienced by Olympia before. Their Information Services Department did not have the necessary skills or resources to conduct the development in house. To combat this, Olympia attempted to enrol an actor with the capabilities to develop Olympia-Online and ensure Olympia's alignment with Horizon. Based on the scripts (imperatives) expressed in the business case documentation, two companies attempted to forge an alignment with Olympia (mediated through the Information Services Architect). Azteka was unsuccessful as they said they were not able to deliver the proposed system within Olympia's desired timeframe (by September 2001). Reflex Technologies was successful as it did promise delivery within the desired timeframe and also offered a fixed-price contract. Developers from Reflex Technologies however developed a prototype based on the Emperor technology – a proprietary rules engine of which they were the sole reseller in Australia – in a very short time period. The prototype played the key role in enrolling Reflex Technologies into the Olympia-Online development actor-network. As a non-human actor the prototype gave an impression that the Emperor was an appropriate technology upon which the new system could be built. The contract signed July 2001 marked the beginning of the Olympia-Online development project which lasted for 12 instead of planned 3 months. This was later on referred to as Phase 1. After a year in operation the development continued mid 2003 until April 2005 as Phase 2. We now describe the Olympia-Online development project as it evolved through the phases.

5.1 Phase 1 Olympia-Online design

Phase 1 development began with some initial requirements-gathering sessions, run by business analysts from Reflex Technologies:

> We ran some formal requirements gathering sessions, first of all more in terms of use case development, just running scenarios to try and understand functionally what the product was supposed to do. (Alan, a Business Analyst)

Once development was underway and Reflex Technologies' developers engaged fully with Olympia's Information Services staff several problems emerged. As the project evolved, Reflex Technologies' developers recognised the actual breadth and depth of the problem at hand. Olympia's internal Information Services staff gradually became aware that Emperor was not the right engine for Olympia-Online's purpose, and was as such misaligned with Olympia-Online's initial objectives. In retrospect, the Senior Information Services Architect involved in commissioning Reflex Technologies noted that Reflex Technologies "didn't understand the problem at hand" and underestimated both its complexity, costs and the required development time. This was seen by the Information Services staff as the major reason for the seven months delay of the project.

The development of Olympia-Online emerged as a complex actor-network. It involved the integration between the existing Mainframe System, the new web-based application on Emperor, an interface with Horizon and PDF documentation development, which would be the customer's final output from the new system. The dynamics of this actor-network reflected the interactions among Reflex Technologies' developers completing the work that required Emperor, and Olympia Information Services staff responsible for all the integration between the Emperor component and the Mainframe resources. In addition, the interface development between the Olympia-Online and Horizon was being carried out by another third party, as was the development of PDF documentation.

That the work was eventually completed suggests that Olympia's interests were inscribed through a succession of translations into the Olympia-Online development actor-network. However the inscription was not strong enough to coordinate and channel the behaviour of actors so as to stabilize the actor-network. The Olympia-Online development actor-network involved eight actors, four of which were actor-networks themselves responsible for specific streams of development. Two other actors were overseeing this work – the Project Manager from Reflex Technologies and the Olympia Head Architect, who in their own words had "trouble ensuring the final system delivered the proposed system's original goals". Although they were officially in charge of Olympia-Online development their association with other actors was not strong enough to 'make others do things' ([9] p.107) and deliver the desired functionality.

Furthermore, GI Business Division, as a powerful actor, put pressure on Information Services. As the original deadline for Phase 1 completion past GI Business Division was anxious to announce to the brokers that the new system is ready for use. They promised that full functionality will be available by mid 2002, which upset Information Services personnel:

> [GI Business Division] shouldn't have gone out and promised that because there's no way in hell we can do it. We just had hundreds of defects outstanding, large parts of functionality not working… At the end of the day they convinced us, everybody put in a huge amount of effort and we sort of got it working with one or two brokers, full functionality, ah, I think somewhere in July [2002]. (George, an Information Services Architect)

However, despite 7 months delay, GI Business Division considered the implementation of Phase 1 Olympia-Online a success. This is based on the positive feedback from the brokers. The success was primarily due to Olympia recognising the power and value in aligning the Company with Telcom and the new Horizon platform. This alignment, combined with Olympia-Online's novelty value, which in turn was associated with being first-to-market with a web-based business insurance product, ensured the new system's success from a market perspective.

During the implementation and operation of Olympia-Online system Information Services staff experienced its poor technical quality, slow performance, frequent crashes and numerous defects. Its design was not modular and hence the system lacked the ability to be scaled to Olympia's future needs. As a result

Information Services staff had huge difficulties in maintaining Olympia-Online. They believed that Olympia-Online failures were caused primarily by the involvement of inappropriate actors – Reflex Technologies and Emperor. Emperor was originally designed for a different purpose and was not able to efficiently translate business rules into the rule engine.

However, Olympia's GI Business Division was unaware of the full extent of the system's technical instability, and thought the Olympia-Online system was an unqualified success. Based on this perceived success, GI Business Division, in discussions with Reflex Technologies, made the decision to purchase $1 million worth of Emperor software licensing such that the existing system could be rebuilt upon and more insurance products could be developed in the future. As Information Services did not enrol GI Business Division into the development actor-network, they were left out of this decision. GI Business Division, on the other hand, relied on their networks with brokers and Reflex Technologies. By purchasing the licence for Emperor they strengthened the actor-network with the Reflex Technologies and – perhaps inadvertently – translated their interests into Olympia-Online development actor-network. That Information Services, the only department that had the technical understanding and expertise, were neither involved nor consulted in this decision, had further implications for Olympia-Online redevelopment.

When the original Olympia-Online system became so unstable that its use could no longer be sustained, GI Business Division ultimately decided to redevelop Olympia-Online. However, since GI Business Division had already spent $1 million on licensing, redevelopment was planned again based on Emperor despite Information Services objections. This was then called Olympia-Online 'Phase 2' beginning mid 2003 and finishing in April 2005 when the system went alive.

5.2 Phase 2 Olympia-Online design

GI Business Division emphasised that their major goal of Phase 2 was to bring Olympia-Online development and knowledge in house, since it was key to Olympia's overall strategy and they wanted to prevent tacit knowledge and expertise from escaping the confines of the Company. This, goal, however, was not achievable since Olympia was reliant on Reflex Technologies' expertise during system development and design, as they were the only knowledge providers for Emperor in the market. Essentially, Emperor had become a delegate for Reflex Technologies' interests, and through the purchase of the Emperor license, Olympia had effectively enrolled itself in Reflex Technologies' actor-network, translating and aligning its own interests with those of the third party provider, as opposed to the other way around.

Phase 2 started with establishment of two new roles: the Business Project Manager responsible for ensuring internal business functionality of Olympia-Online and the IS Project Manager responsible for project completion on time and within budget. The third important actor was the Broker Business Experts (from GI) who presented the Broker Community's views. The Broker Business Expert managed to wield considerable influence on Olympia-Online development by translating the project objectives to be aligned with his own. This influence ensured the Phase 2 system was not implemented until a sufficient level of Broker Community functionality and quality had been delivered. He also made sure that the new system would be superior to both the existing system and other web-based products that competitors had recently developed in an attempt to attain parity with Olympia-Online.

This Broker Business Expert's high level of involvement resulted in the inscription of the Broker Community's interests in the new system, a strong alignment between the system and the Broker Community, and Olympia-Online's continued market success. This success in the market, however, is once again contrasted with internal failures.

The Business Project Manager appeared to be rather aloof and didn't engage in system development. As a result business requirements, including internally needed administrative functionality such as management and operational reporting, were not included in the Phase 2 Olympia-Online design. This means that GI's objectives were not inscribed in the new system.

Furthermore, the IS Project Manager's focus on time-line and budget control led to Olympia-Online design that was non-scaleable. Such a design would not enable adding easily new insurance products in the future. From this perspective the GI Business Division's interests were not aligned with or enrolled in the new Phase 2 development actor-network. This outcome is particularly disappointing for GI representatives, as they were promised the delivery of such functionality. A Senior Manager from within the GI Business Division expresses this frustration:

> [Phase 2] should have been *it*. So, well, you spend a considerable amount of money on Phase 1, you get to redo it in Phase 2 and it's disappointing when you hear you might have to do it again in Phase 3, to get what you actually thought you'd be getting in Phase 2.

This absence of core internal functionality has distanced the GI Business Division even further from the Olympia-Online development actor-network. Because they didn't see their interests inscribed they disaligned themselves from this actor-network. This might have long-term consequences for Olympia as further development of Olympia-Online was not planned while their competitors in the industry were building comparative systems.

6 Discussion and Conclusion

The ANT analysis of the strategic IS Olympia-Online development project reveals an open-ended structure of heterogeneous actor-networks which provide a rich description of its socio-technical nature. Such a description enables unpacking of the socio-technical interplay between the project actors, both human and non-human, which in turn explains how both success and failure of this strategic IS development were constructed. The key to success of Olympia-Online development was the ability of Information Services staff to align diverse interests of Olympia, the Broker Community and the new Telcom system Horizon and inscribe these interests into the Olympia-Online development actor-network. Especially by translating brokers' needs and interests into the design of Olympia-Online, or in other words, by inscribing these interests in 'durable materials' (Law, 1992), Olympia-Online implementation and subsequent use strengthened and stabilized this actor-network.

On the other hand, persistent technical problems faced during the development and the failure of Olympia-Online to deliver internal business functionality can be seen as resulting primarily from GI Business Division's simultaneous weak alignment with their own department of Information Services and strong alignment with Reflex Technology. Firstly, by enrolling Reflex Technology into the Phase 1 development actor-network GI Business Division enabled them to inscribe their interests through a succession of translations into Olympia-Online via their proprietary technology Emperor. Secondly, such inscription was highly strengthened by GI Business Division's decision to purchase the Emperor software licence. Alternatively, GI Business Division could have followed Information Services' recommendations not to continue further Olympia-Online development based on the Emperor engine. The concept of irreversibility of an aligned actor-network explains the impact of such a decision (Callon, 1991). As we have seen, purchasing of the Emperor software licence shaped and determined subsequent translations in the Olympia-Online development actor-network and caused significant technical problems and perceptions of the system technical failure with long-term implications. This decision produced irreversibility of this actor-network as it became "impossible to go back to a point where that translation was only one amongst others" (Hanseth and Monteiro, 1998, p. 100). The longer Olympia-Online development continued on Emperor technological platform the degree of irreversibility of its actor-network became higher. As a result technical problems persisted and complexity expanded taking more time and resources. While increasingly misaligned with Olympia-Online development actor-network GI Business Division perceived the project as failure from a project management perspective (over time and budget) and for not delivering the desired internal functionality.

By following the actors – human and nonhuman – and by tracing their associations as they created actor-networks of the strategic IS development project we described simultaneous making of its success and failure. Through a historical reconstruction of this project, we traced the emergence of heterogeneous actor-networks and revealed how and why some actors succeeded in translating their interests into the IS designs while others didn't. These processes of actor enrolments, translations and inscriptions of particular interests led to instability of some heterogeneous actor-networks and strengthening others, thereby producing the perceptions of failure and success.

References

1. Aanestad, M. (2003) The camera as an actor: design-in-use of telemedicine infrastructure in surger. Computer Supported Cooperative Work 12: 1-20
2. Avison, D., Jones, J., Powell, P. and Wilson, D. (2004) Using and validating the strategic alignment model. Journal of Strategic Information Systems 13: 223-246
3. Bloomfield, B.P., Coombs, R., Cooper, D.J. and Rea, D., (1992) Machines and manoeuvres: responsibility accounting and the construction of hospital information systems. Accounting, Management and Information Technology 2,4: 197-219
4. Callon, M. (1991) Techno-economic network and irreversibility. In: Law J. (ed) A Sociology of Monsters. Essays on Power, Technology and Domination, Routlegde, pp 132-164
5. Cavaye, A.L.M. and Cragg, P.B. (1993) Strategic information systems research: a review and research framework. Journal of Strategic Information Systems 2(2): 125-137
6. Doolin, B. and Lowe, A. (2002) To reveal is to critique: actor-network theory and critical information systems research'. Journal of Information Technology 17: 69-78
7. Hanseth, O. and Monteiro, E. (1998) Understanding Information Infrastructure (manuscript), http://heim.ifi.uio.no/~oleha/publications (accessed 18.04.06)
8. Hendry, J. (2000) Strategic decision making, discourse, and strategy as social practice. Journal of Management Studies 37,7: 955-977
9. Latour, B. (1991) Technology is society made durable. In: Law J. (Ed.) A Sociology of Monsters. Essays on Power, Technology and Domination, Routlegde, pp 103-131
10. Latour, B. (2005) Reassembling the Social: An Introduction to Actor-Network Theory, Oxford University Press
11. Law, J. (1992) Notes on the theory of the actor-network: ordering, strategy and heterogeneity. Systems Practice, 5, 4: 379-393
12. Luna-Reyes, L.F., Zhang, J., Gil-García, J.R. and Cresswell, A.M. (2005) Information systems development as emergent socio-technical change: a practice approach. European Journal of Information Systems 14: 93-105
13. Luftman, J. and Brier, T. (1999) Achieving and sustaining business-IT alignment. California Management Review 24,1: 109-122
14. Mitev, N.N. (1996) More than a failure? The computerized reservation systems at French Railways. Information Technology and People 9,40: 8-19
15. Myers, M. D. (1994) A disaster for everyone to see: an interpretive analysis of a failed IS project. Accounting, Management, and Information Technologies 4,4: 185-201

16. Neuman, W.L. (2006) Social Research Methods—Qualitative and Quantitative Approaches, 6th ed., Allyn and Bacon, Boston
17. Reich, B.H. and Benbasat, I. (2000) Factors that influence the social dimension of alignment between business and information technology objectives. MIS Quarterly 24,1: 81-113
18. Tatnall, A., and Gilding, A. (1999) Actor-network theory and information systems research. Proceedings of the 10th Australasian Conference on Information Systems, pp. 955-966
19. Underwood, J. (1998) Not another methodology: what ANT tells us about systems development, The 6th International Conference on Information Systems Methodologies, British Computer Society, Salford UK, accessed from http://www-staff.mcs.uts.edu.au/~jim/papers/ismeth.htm
20. Underwood, J. (2001). Negotiating the chasm in system development : some advice from ANT, Proceedings of the 12th Australasian Conference on Information Systems Coffs Harbour, NSW
21. Vidgen, R. and McMaster, T. (1997) Black boxes: non-human stakeholders and the translation of IT through mediation, Proceedings of the IFIP TC8 WG 8.2 International Conference on Information Systems and Qualitative Research, pp. 250-271
22. Walsham, G. (1995) The Emergence of interpretivism in IS research. Information Systems Research 6,4: 376-394
23. Walsham, G. (1997) Actor-network theory and IS research: current status and future prospects, Proceedings of the IFIP TC8 WG 8.2 International Conference on Information Systems and Qualitative Research, pp. 466-480
24. Walsham, G. and Sahay, S. (1999) GIS for district-level administration in India: problems and opportunities. MIS Quarterly 23,1: 39-65

How Can Organizations Achieve Competitive Advantages Using ERP Systems Through Managerial Processes?

Carl Erik Moe, Erik Fosser, Ole Henrik Leister, Mike Newman

Agder University College, Servicebox 422, 4604, Kristiansand, Norway

Abstract: Enterprise Resource Planning (ERP) systems have become a standard for collecting all business areas in one system. For some organizations it is a necessity for doing business, while others use it to outperform competitors. Prior research in the area of ERP systems and competitive advantage is dispersed and insufficient. Based on literature review and in-depth interviews at two Norwegian enterprises we have developed a framework which we believe gives insight into the areas managers should focus on when implementing ERP systems.

Our study supports the claim that ERP systems in themselves do not create a competitive advantage. Instead, they form an information basis which managers can utilize to outperform competitors. A good relationship between IT/IS managers and the rest of the management is crucial. The importance of top management support was highlighted and linking ERP and a data warehouse as a "bolt-on" was very beneficial in one of the enterprises.

1 Introduction

Enterprise Resource Planning (ERP) systems' aim to integrate all business processes into one enterprise wide solution by having one centralized database that all business areas have access to [21]. The integration facilitates information flow throughout the entire company [22]. It is possible to customize the ERP system to fit the original business processes. However this is usually not recommended because of the high cost and problems with system upgrades and maintenance difficulties [12]. Benefits from using ERP-systems are said to be revolutionizing [10]. However as our literature review shows, there are also stories of failures, cost overruns and problems with change management, hence it is of vital importance to understand how to achieve the promised benefits.

2 Prior research

Lengnick-Hall et al. [16] claims that ERP systems do not offer competitive advantages in themselves, but has to be coupled with social and intellectual capital within the firm. This is supported by others [14; 20; 23]. The overall benefit lies in the quality of *integrating* an organization's information flow and business processes [5]. By doing this the enterprise can optimize resources. However, others have argued that ERP can be part of making a competitive advantage in some situations [3].

Drawbacks with ERP systems have also been reported. Managers have not been aware of the difficulties in changing processes and organizational culture to fit the ERP system and struggled with change management [21; 18]. ERP systems may eliminate competitive advantages that organizations possessed before the implementation [3], due to the "common systems approach". Future needs of the enterprise can be hard to predict and further development of the ERP system is unknown to the buyer [18]. Drawbacks with lock-in of an organizations processes and principles into a specific software solution have been reported [15]. Shortcomings may lead to extensive switching costs, and it can be costly to combine the ERP system with other software products [22].

Not much research has been done on managerial processes concerning ERP systems and creating a competitive advantage [20; 3; 13]. More research has been done on the topic of implementation and realizing the benefits of ERP systems [9; 13]. A few studies has focused on mature systems up and running and achieving competitive advantage. This era has been called the "Post-Net" era [3] and "second wave" [6].

Kalling [14] focuses on how ERP systems and strategic management processes can lead to a competitive advantage. He suggests a framework to improve understanding of processes to achieve competitive advantage and uses the concept "bricolage" [7]. By this he means learning through trial and error and local tinkering. He argues that this may lead to improved work practices, new capabilities and therefore a strategic advantage. Many studies argue that it is important to have an open environment built on trust [16; 23]. The innovative organization which lies on top of the mechanistic ERP system should focus on open communication, consensus, alignment and flexibility [23]. The findings are dispersed and have sprung out of different school of thoughts. Most have never been tested empirically.

To structure our research, we developed a framework based on previous findings, shown in table 1, below. In our paper the resource-based view is used to define competitive advantage, the definition of resources [28] including broad aspects as brand names, in-house knowledge of technology, employment of skilled personnel etc. Mata et al. [20] concluded that managerial IT skills is the only attribute expected to create a competitive advantage. Powell and Dent-Micallef [23] found that flexible culture and supplier relationships are important. The results support the resource-based approach, and help to explain why some firms outperform others using the same ITs.

Table 1. Overview of significant prior research

Research domain	Our Research Topic	Prior Research
General	Common System Paradox	[3; 18; 16]
Competitive advantage	ERP systems and comp. advantage	[16; 3]
	The organizations' comp. advantage	[16; 3; 13; 14; 2]
	The managers' comp. advantage processes	[20; 1; 23; 16; 14]
Foundation for competitive advantage	The managers' knowledge of the organization and the ERP system	[1; 16]
	Training	Own curiosity
	Communication	[23; 20; 16]
	Cultures	[23; 15; 25; 21; 2]
	Business competent IT/IS dept.	[20; 13; 17]
	Top management support	[23; 22]
	Organizational structures and processes	[23; 16; 13; 25]
	Extraction of information	[16; 8]
	Creative usage of information	[16; 8]
	Extension of the ERP system	[26; 24; 4; 27; 19]
Processes	Future focus areas	[13; 14; 1]
	Plans for achieving comp. advantage	[14]
	Hurdles to achieve comp. advantage	[18; 14; 22; 21]

Based on the literature review we expect that ERP-systems can lead to competitive advantages, but that this depends on specific foundations in the organisation and on specific managerial processes. Our research question is therefore "What foundations and managerial processes can lead to competitive advantages when using ERP-systems?" To answer this we collected data at two organisations that were using the same ERP-system, and we will in the following briefly present our research approach before we present the results.

3 Research approach

We carried out 16 semi-structured interviews at the two organizations locations. As part of our triangulation we did an analysis of their IT/IS-strategies. The interviewees were asked to read through the "Interpretation and analysis" and the "Discussion" chapter for confirmation of our findings. Both companies are in the food industry. Foody had revenue of 485 millions NOK in 2004 and an IT budget of 4.5 million. Sllims revenue was 1.5 billions, and the IT budget was as much as 27.5 million. Foody had one department dealing with technology with only 3 employees, led by the director of finance. Sllim has both an IT and an IS department, both with their own manager.

4 Results and analysis

The chapter reports our main findings; table 2 below gives an overview.

4.1 General

All but one of the managers at Sllim said they were satisfied or very satisfied with their ERP system. The unsatisfied manager claimed that the system gave little support to his department's processes. At Foody five of the seven interviewed were satisfied with the ERP system. One was not satisfied and expressed it this way:

> *"The system is too hard to work with and navigate in. .... It is an A4 system, which force us to change to fit the system."*

The biggest benefit at both organisations was integration of data. This enabled them to streamline their value chain and internal processes across the entire business. The biggest drawback was that it was a bit too complex. Other drawbacks mentioned were that the sales module was not good enough, the system was too technical, and a lot of skills were needed to both use and manage it.

4.2 Competitive Advantage

At Sllim competitive advantage was something that had been discussed right from the start. Almost everyone stated that an ERP system could only give competitive advantage together with other systems or special competence. The director of IS explained that if their goal only was to maintain the status quo, they would be better off without the system. But when they added the data warehouse, they achieved an enormous data basis, which could give competitive advantage. At Foody they had difficulties in seeing potential for competitive advantage. The only two that really saw the advantage of the ERP system was the CIO and the director of finance. They explained what benefits the system could give, but had less focus on how they could differentiate themselves.

We wanted to know what different managers did to achieve a competitive advantage with basis in the system. We were especially interested in how they were able to arrange information, define structures and processes, customise the system, and manage employees. At Foody they mostly talked about improving. The answers we got at Sllim were reflective, but quite varied because of different areas of responsibility. The director of finance stated:

> *"I try to exploit the available tools in SAP, without investing money in new functionality. There are a lot of possibilities in the ERP systems, e.g. HR,... to utilize our resources more efficiently."*

Other managers at Sllim focused on a tight cooperation with the business and tried to develop the business processes. Others again worked with parameter settings to optimize the system. The organization focused continually on training to enhance the user skills. They were also doing projects with suppliers and customers to extend their value chain. There was clearly focus on how to continuously develop and improve their work routines.

4.3 Foundation for Competitive Advantage

We wanted to know if the knowledge of the organization and the ERP system combined could be a source of competitive advantage. The two organizations are fairly different in this respect. Sllim rely on their IS department to improve their processes and to be innovative. The knowledge about both the organization and the ERP system are at its greatest in this department. The employees had worked in the company for a long time and knew the organization and the ERP system well.

At Foody they focused on the knowledge about the system and not the organization. Some thought they were finished with the implementation. The need for continual improvements, adjustments and the managers need for someone to push them were highlighted. The managing director confirmed that the combination of knowledge about the organization and the ERP system could be a source of competitive advantage. At the same time he was not very supportive concerning this.

Business competent IT/IS department is believed to be important. Sllim had a well established IS department with high focus on business knowledge as well as systems development. Dividing the IT and IS department helped them to avoid technical issues and focus on business development. The functional managers overwhelming trust in the IT/IS department impressed us. Their knowledge about the enterprise's processes and their strong determination to solve problems made them very effective. The IS department had tight communication with the users. Structured interviews to gather users opinions, was done once a year. Less structured communication was done continuously by walking around. At Foody the managers were divided concerning this issue. One indicated that their IT department had too much focus on technology instead of information systems as a whole. Another said that the IT department's knowledge was insufficient about his department's routines. However, the majority had a very different view.

We asked the individual managers how and if they encouraged their employees to utilize the information in the ERP system. The answers at Sllim were a bit diverse. Many reported that in some way they focused on extracting information and that the link to the data warehouse was very important. The data warehouse helped the employees to get structured reports that were used to further develop the organization. The director of IS explained that they focused on best practice and tried to transfer new, innovative knowledge to other departments. However, the ERP system was not set up to use information creatively. The job of the system was to deliver information to their data warehouse where all analysis was done.

Foody was concerned with how to extract information from the ERP system. When new ways of using the system was revealed, manuals were distributed to the various departments. The director of finance and other members of the original implementation team often developed intuitive user guides to help the employees get more out of the system. The director of production said they had a long way to go to use information creatively. Others argued that they were now in a “phase two” and could focus more on benefits.

The two organizations are at different levels, Sllim has had their ERP system for some time. They have created their own data warehouse which enables them to do comprehensive analysis and find creative improvements. Foody has not yet managed to stabilize their ERP system to develop their analytical skills. This topic of extension of the ERP system was mentioned by different managers during the interviews, at both enterprises. The bolt-ons could produce more firm specific advantages if configured properly. Sllim has had success with their data warehouse, whereas at Foody the managing director stated that *“we should probably have a data warehouse”*.

4.4 Processes

There had been limited focus on pre-analysis in both organizations. Still, Sllim had done more. The managers had some examples on what they did in the pre-phase of implementation. Not everything was done in the context of competitive advantage, however. Interviews were conducted to detect employees’ need for support, and a number of requirements were identified from the entire organization. By tuning the system to the organization and the other way around, they achieved a good fit between system and organization. They had a “Phase II” of the implementation where they focused on how to exploit the extra efficiency gains of the system.

We wanted to know if the organizations had any plans or strategies on how competitive advantage could be achieved and if these were followed up and evaluated. This topic was discussed to identify their systematic approach to achieving a competitive advantage. It seemed that both organizations used their business strategies as pointers for what projects and processes they would initiate. They had however troubles with relating this to competitive advantage.

5 Discussion

How mature and stable the ERP system is, has impact on the level of utilization [22]. Markus and Tanis’s [18] framework describes four phases in an ERP systems life cycle; the chartering phase, the project phase, the shakedown phase, and the onward and upward phase. Foody is still in the “shakedown” phase, whereas Sllim is in the “onward and upward” phase. The fact that the two enterprises were in different phases and that Sllim has been running their ERP-system for two more years than Foody has to be considered when we discuss the findings. But it was still a paradox that Foody had spent so little time on how the ERP system could give them a competitive advantage. Managers at Sllim have a more comprehensive and clearer strategy on how to succeed. This could also affect their level of adoption. The data warehouse gave them a competitive advantage and the ERP system was a part of this configuration. This is in line with [19] concerning data warehousing and [3; 14; 16] on information utilisation.

ERP implementations [14; 16] may eliminate competitive advantages an or-organization has. Features that made them unique and hard to imitate may be destroyed because of the adjustments to the system. At Foody, one situation may illustrate this effect. The director of production preferred an old customised system, it gave him all the information he needed and it was specially designed to fit his department's processes. Others told us that they were good at production, which confirms that the old system may be better suited for that department.

Mata et al. [20] described four important managerial skills to achieve a competitive advantage. These are IT managers' ability to: 1) understand business needs, 2) cooperate with other managers, 3) coordinate IT activities, and 4) anticipate future IT needs. Three of these skills were identified at Sllim, only one at Foody. We did not find evidence for the dual core structure, but it could be that it is used unconsciously. The dual core structure consists of an operational technical core and a strategic learning core [16]. At Sllim, they had opened for cross functional training. Users feeling they would benefit from extending their knowledge about surrounding departments' ERP processes could freely join. This can be seen as a path towards a dual core structure.

In both organizations the IT/IS department worked as a driver for ERP implementation and utilisation. The CIO and the director of finance at Foody were capable of being change agents. At Sllim, the IS department was really driving all other departments forward. One manager there told us that their competitive advantage consisted of an organizational culture compound with the ERP system which was not possible to copy, but he could not put words to it. It could be that he talked about a strategic important resource that fell under two of the conditions developed by Barney [2]; causal ambiguity and social complexity. This resource might be a tacit capability which was intertwined in social network and relationships [2].

Powell and Dent-Micallef [23] argues that supplier relations can create a competitive advantage. Managers at both companies talked about the importance of extending their value chains. One of the main reasons for implementing the ERP system at Foody was the possibility to exchange information with their biggest customer.

6 Summing up

Our study has confirmed some of the concepts from prior research. We confirmed that ERP systems do not give competitive advantage on their own, but can be the information basis for achieving an advantage. The few managerial skills mentioned in the literature have also been confirmed. What stood out as a beacon is the importance of a good relationship between the IT/IS manager and the rest of the management. The importance of top management support was confirmed. The management and organizational culture, and good supplier relations were also confirmed to some degree. The framework we developed can be important for further research. Below is a modified version of Table 1 with a summary of results and an overview of our contribution.

Table 2. Framework and findings

Research	Domain	Our Research Topic Support
General	Common System Paradox	No support
Competitive advantage	ERP systems and competitive advantage	ERP systems ability to create a competitive advantage had strong support in one
	The organizations' competitive advantage	organization and only weak support in the other
	The managers' competitive advantage processes	No evidence indicating that managers had a structured process
	The managers' knowledge of the organization and the ERP	Weak support for the learning loops described in the prior research, but it could be incorporated and hidden
	Training	in training.
	Communication	Opinions were highly dispersed and ranged from no support to strong support.
Foundation for competitive advantage	Cultures	Only a few highlighted the importance, but these few were strongly convinced.
	Business competent IT/IS department	Strong support
	Top management support	Strong support
	Organizational structures and processes	No or very weak support for the concept of the dual
	Extraction of information	core structure in both
	Creative usage of information	enterprises.
	Extension of the ERP system	Strong support for the importance of bolt-ons.
	Pre analysis	Some support for the importance of pre-analysis.
Processes	Future focus areas.	None of the organization had structured plans.

Research	Domain	Our Research Topic Support
	Plans for achieving competitive advantage	Both had focus areas and projects, but the link was not always visible.
	Hurdles to achieve competitive advantage	Hurdles described in prior research have been supported.

References

1. Andreu R, Ciborra C (1996) Organisational Learning and Core Capabilities Development: The Role of IT. J Strat Inf Sys. 5:2, 111-127
2. Barney JB (1991) Firm Resources and Sustained Competitive Advantage. J of Man 17:1, 99-120
3. Beard JW, Sumner M (2004) Seeking Strategic Advantage in the Post-Net Era: Viewing ERP Systems from the Resource-Based Perspective. J Strat Inf Sys 13:2, 129-150
4. Bendoly E, Kaefer F (2004) Business Technology Complementarities: Impacts of the Presence and Strategic Timing of ERP on B2B E-commerce Technologies Efficiencies. Int J Man Sci 32:5, 395-405
5. Brady JA, Monk EF, Wagner BJ (2001). Concepts in Enterprise Resource Planning. Thompson Learning, Boston
6. Brown CV, Vessey I (2003) Managing the Next Wave of Enterprise Systems: Leveraging Lessons from ERP. MIS Q Exec 2:1, 65-77
7. Ciborra C, Jelassi T (1994) Strategic Information Systems: A European Perspective. John Wiley and Sons Ltd., Chichester.
8. Daft RL (1978). A Dual Core Model of Organisational Innovation. Acad of Man J 21:2, 193-210
9. Davenport TH (1998) Putting the Enterprise Into the Enterprise System. HBR 76:4, 121-131
10. Davenport TH, Harris JG, Cantrell S (2002) The Return of Enterprise Solutions: The Director's Cut. Retr Oct 1st 2004, from http://www.accenture.com/xd/xd.asp?it=enwebandxd=_ins%5Cresearchreportabstract_174.xml.
11. Holland C, Light B, Kawalek, P (1999) Beyond Enterprise Resource Planning Projects: Innovative Strategies for Competitive Advantage. Proc 7th Eur Conf Inf Sys 1: 288-301
12. Kalling T (1999) Gaining Competitive Advantage through Information Technology: A Resource-Based Approach to the Creation and Employment of Strategic IT Resources. Lund Business Press, Lund
13. Kalling T (2003) ERP Systems and the Strategic Management Processes that Lead to Competitive Advantage. Inf Res Man J 16:4, 46-67

14. Kumar V, Maheshwari B, Kumar U (2003). An Investigation of Critical Management Issues in ERP Implementation: Empirical Evidence from Canadian Organizations. Technovation 23:10, 793-807
15. Lengnick-Hall CA, Lengnick-Hall ML, Abdinnour-Helm S (2004). The Role of Social and Intellectual Capital in Achieving Competitive Advantage through Enterprise Resource Planning (ERP) Systems. J Eng and Tech Man 21:4, 307-330
16. Markus ML, Benjamin RI (1996). Change Agentry – the Next IS Frontier. MIS Q 20:4, 385-407
17. Markus ML, Tanis C (2000). The Enterprise Systems Experience – From Adoption to Success, Zmud, R.W. (Ed.) Framing the Domains of IT Research: Glimpsing the Future through the Past. Pinnaflex Ed. Resources, Cincinnati.
18. Martin J (2000) A Fast Fix. Manufacturing Sys (MSI), 18:3, 88-94
19. Mata FJ, Fuerst WL, Barney JB (1995) Information Technology and Sustaining Competitive Advantage: A Resource-based Analysis. MIS Q 19:4, 487-505
20. O'Leary DE (2002) Enterprise Resource Planning Systems: Systems, Life Cycle, Electronic Commerce and Risk. Cambridge Univ. Press, Cambridge.
21. earlson KE, Saunders CS (2004) Managing and Using Information Systems (2nd ed.). Leyh Publishing LLC, New Caledonia.
22. Powell TC, Dent-Micallef A (1997) Information Technology as a Competitive advantage: The Role of Human, Business and Technology Resources. Strat Man J 18:5, 375-405
23. Rich N, Hines P (1997). Supply Chain Management and Time Based Competition: The Role of the Supplier Association. Int J Distr and Log Man 27:3-4, 210-225
24. Somers TM, Nelson KG (2003). The Impact of Strategy and Integration Mechanisms on Enterprise Systems Value: Empirical Evidence from Manufacturing Firms. Eur J Op Res 146:2, 215-228
25. Unal A (2000) Electronic Commerce and Multi-Enterprise Supply/Value/Business Chains. Inf Sci 127:1-2, 63-68
26. Vassiliadis P, Quix C, Vassiliou Y, Jarke M (2001) Data Warehouse Process. Man Inf Sys 26:3, 205-236
27. Wernerfelt B (1984) A Resource-based View of the Firm. Strat Man J 5:2, 171-180

Novel Approach to BCG Analysis in the Context of ERP System Implementation

Neven Vrček, Željko Dobrović, Dragutin Kermek

University of Zagreb, Faculty of Organization and Informatics, Pavlinska 2, Varaždin 42000, Croatia

1 Introduction

ERP systems have been the topic of interest for the researchers and practitioners for several decades. There are many success stories about their use and implementation [2, 42, 47], but also there are significant evidences of failures [16, 17, 18, 21, 35, 48, 49]. The controversy about them mostly arises from the fact that extremely high costs of licenses and implementation are not clearly related to the benefits of the enterprise [13, 15, 26, 28, 31, 45]. Also the measure of enterprise benefit has been vaguely defined with various performance indicators ranging from purely financial to complex composite performance measures [3, 14, 25, 29, 33, 34, 27, 37, 38, 41, 43, 46]. The reasons for low achievement of the desired goals have been found in various factors during implementation procedure, and there are several recommendations for better implementation approach (4, 6, 7, 50). It is generally accepted that every implementation requires strong organizational and management commitment (8, 20).

Standard ERP systems have been promoted as software solutions that are built on best business practices. According to the promoters, enterprises that buy them should embrace those built in business templates and achieve excellent business results. However, if enterprise has specific needs which are not part of standard ERP functionality, long lasting, expensive and complicated procedure of customization has to be carried out [5, 23, 24].

The link between competitive advantage and information system has been strong motivator for managers to buy ERP systems and force their implementation. However it is intuitively clear, that if something is commercially available to broad customer base, it cannot bring significant competitive advantage [36]. Many authors recognized that, and although there are certain positive indicators of business performance related to standard ERP systems, there is no strong evidence of such relationship [19, 44]. So what an enterprise or a non-profit organization can actually expect from an ERP system, and what benefits does it bring? It seems that certain organizations are better suited for ERP system implementation than the others [1, 30, 31].

Our results show that every serious attempt, which tries to answer this question, has to dig deep into business processes and find out what is their operational and strategic significance. Such comprehensive analysis requires significant time and strict methodology that enables unambiguous comparison between various organizations. This paper presents results of such approach. It is based on several years of research in various organizations that implemented or considered implementation of an ERP system. For that purpose authors developed methodology for strategic planning of information systems (SPIS). It was published in several scientific and professional articles and verified in relatively large number of commercial projects [10, 11, 12]. The methodology consists of a significant number of well known methods, but it combines them into structured and chained holistic process which gives deep insight into the business system and corresponding information system. By combining various methods under a common framework of methodology we obtain detailed and documented picture of an organization.

Table 1. Steps of SPIS methodology

Problem/step in IS design	*Methods and techniques* § -*strategic*, # -structured, ¤ -object oriented	**Inputs and deliverables** Inputs / Outputs	*Usability* **V**ery powerful **P**owerful, **U**seful
1. Description of Business System (BS)	Interviewing	*Missions and goals of current BS* / Business strategy; Business processes (BP)	
2. Evaluation of the Impact of New IT on Business System	§ Balanced Scorecard § BCG-matrix § 5F-model § Value-chain model	*BP* / Performances of existing BS *Business strategy* / IS development priorities *Business strategy* / Information for top management *BP* / Basic (primary and support) business processes (BBP)	V *P* U V
3. Redefinition of Business Processes	# BSP-decomposition # Life cycle analysis for the resources	*BBP* / New organizational units (OU) *Basic system resources* / Business processes portfolio	P P
4. Business System Reengineering	§ BPR § SWOT	*Business Processes Portfolio* / New business processes (NBP) *Business Processes Portfolio* / SWOT analysis for NBP	P V
5. Estimation of Critical Information	§ CFS analysis (Rockart) # Ends-Means analysis	*NBP* / Critical information for NBP *NBP* / Information for efficiency and effectivity improvement	P U
6. Optimization of New IS Architecture	# Matrix processes entities # Affinity analysis, Genetic algorithms	NBP / Business process relationships *Business processes relationships* / Clusters; Subsystems of IS	V *P*
7. Modeling of New "Business Technology" (BT)	# Work flow diagram (WFD) # Organizational flow diagram (OFD) # Activity flow diagram (AFD)	*NBP* / Responsibility for NBP *New OU* / Flows between new OU *NBP* / Activities for NBP	V P U
8. Modeling of New Business Processes, Supported by IT	# Data flow diagram (DFD) # Action diagram (AD)	*NBP* / NBP supported by IT (IS processes); Data flows; Business Data *IS processes* / Internal logic of IS processes	V P
9. Evaluation of New IS Effects	# Simulation modeling	*IS processes* / Guidelines for BP improvements	U
10. Business Data Modeling	# ERA-model ¤ Object-model	*Business Data* / ERA model *Business data* / Objects model	V P
11. Software Design	# HIPO- diagram ¤ Transition diagram	*IS processes* / Logical design of program procedures (SW) *Data flows* / Events and transactions	V P
12. Detail Design of Programs and Procedures	# Action diagram ¤ Object scenario	*Logical design of program procedures (SW)* / Model of program logic *Object model; Events and Transactions* / Objects behavior	P P
13. Data Model Development	# Relational model; Normalization	*ERA model* / Relational model	V
14. Software Development	# CASE tools and 4GL ¤ OO-CASE tools	*Model of program logic; Relational model* / Programs and procedures *Object behavior* / OO procedures	P P
15. Implementation of New IS	Case-study; Business games	*Programs and procedures* / Performances of new IS	P
16. Evaluation of New BS Performances	# Balanced scorecard	*Performances of existing BS; Performances of new IS* / Measure of success	*V*

The SPIS has a number of methods, but some are applicable only in the case of the entirely new development cycle that brings new information system architecture. In the case of standard of the shelf software solutions, the methodology has been modified to produce combined information system that will be composed of standard and customized modules. One of the methods, which derived from such approach and proved extremely useful, is a combination of the Boston Consulting Group (BCG) matrix [39, 40] used in step 2 of the methodology, and the gap analysis performed during ERP system implementation. It shows the relationship between strategic relevance of the business processes and the level of support that is going to be achieved by standard software solution. Obtained results prove that in many cases strategic business processes are not supported at adequate level, and gained business results are sub-optimal [22]. Achieved outcomes are systemized and confirm initial hypothesis that standard ERP systems present low level of support for strategic business processes. Without detailed rethinking of the entire business paradigm, supported by really holistic and integral information system, competitive advantage will never be achieved.

2 Standard ERP system implementation methods

ERP vendors usually strongly promote their own approach to implementation of their products. Such examples are ORACLE AIM (Application Implementation Methodology) and SAP ASAP (Accelerated SAP). These implementation concepts have several advantages. They are:

1. Strictly focused and goal oriented to implementation of particular ERP system;
2. Based on standard documentation which enables easy transfer of knowledge and benchmarking of implementation projects;
3. Known by various ERP system consultants, which reduces dependence on certain implementation expert(s);
4. Verified on number of projects, which reduces the risk of successful implementation.

Therefore it is quite understandable that many enterprises, entering high risk and long lasting implementation projects, accept proposed implementation method without deep analysis how it applies to their own circumstances.

However standard implementation methods have serious downsides. Due to the fact that they are focused on software implementation and not on business improvement, they leave very little room for serious rethinking of business technology and business process reengineering. On the course of implementation, changes in business processes are usually proposed as a consequence of disproportion in functionality between ERP system and present work practice. Such forced business process reengineering adapts the enterprise to the ERP system without taking much care about actual business needs.

Standard step in every ERP implementation method is a gap analysis. Gap analysis denotes disproportions between current business processes in the organization and standard ERP system functionality. These disproportions are called gaps and might be solved in various manners, but there is usually very little attention paid to strategic importance of the declared gaps. Usually gap analysis is used to eliminate most obvious problems in implementation projects, and serious gaps, such as complete lack of support for certain business processes, are left to some subsequent projects. However results of the gap analysis are valuable source of information on which strategic impact of ERP system might be estimated.

3 BCG Matrix and Gap analysis

SPIS methodology is extremely detailed and requires strict modeling with description of business processes and data flows, before and after business process reengineering. Business processes are modeled through workflow, and activity diagrams. Workflow diagram presents distribution of processes within organizational units and data flows among them. Activity diagrams decompose each business process on activities performed by various jobs within organization. The consistency of the model is checked by the matrix of business processes and data classes. This matrix is entry value for affinity analysis algorithm, which gives logical grouping of processes with optimal modules and granularity of information system. In the case of standard ERP system, results of affinity analysis might be an early indicator of implementation problems. If grouping of business processes is different than grouping of standard ERP system modules, the level of required modifications might be significant.

At the final stage the model must be accepted by the management together with the plan for implementation of reengineered business technology with the new architecture of information system.

The early step of the methodology is the BCG analysis. This strategic method is important tool for valuating business processes in the context of their dependence regarding information technology and assessing the strategic significance of an information system. It is based on two-dimensional matrix showing four different types of management environments (see Figure 1).

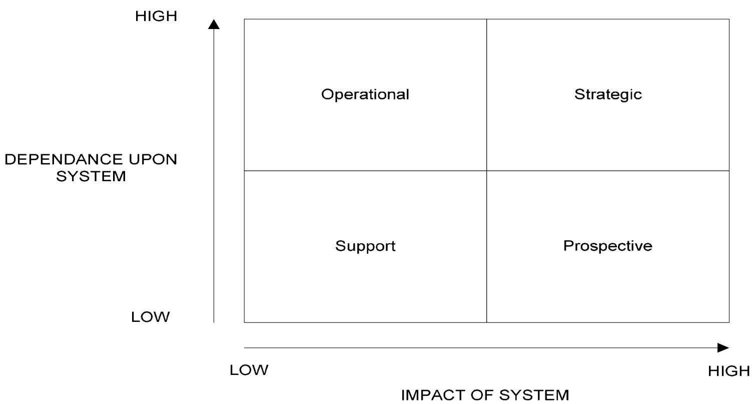


Fig. 1. Framework for assessment the strategic importance of an IT system

During analysis business processes are classified according to BCG matrix. Various business entities have different classifications of business processes according to the business sector or industry in which they operate and their strategic incentives. It is quite obvious that competitive advantage can be achieved by market differentiation which is obtained by excellence in strategic business processes. These processes must be supported by the information system to reach and maintain competitive edge. Therefore it is interesting to analyze how commercial of the shelf software packages relate to these processes and hence to the competitive advantage.

New in SPIS methodology is that, in this early step, results of BCG analysis are related to gap analysis. Gap analysis is used to analyze the level of support which will be given to the business processes by the standard functionality of ERP system. This analysis can not be performed on generic level but requires deep understanding of real ERP system functionality.

All ERP systems are developed on certain business templates and they have problems in situations where these templates are not suited to real organization which implements such standard software solution. These gaps might be solved by the change in the software, organization or, quite often, not at all. The question is how the existence of gaps relates to the competitive advantage of the enterprise and how are business processes that differentiate enterprise on the market supported by ERP system. This paper tries to find an answer by correlating BCG and gap analysis.

4 Materials and methods

The research was conducted in the period from 1996 to 2004. During that period the SPIS methodology was applied to nine large-scale projects. The projects, from which results presented in this paper were derived, took place in:

1. **Big international food industry (p1);**
   The company decided to improve business performance by introducing the ERP system. The decision for choice of an ERP vendor was based on benchmarking with similar industries on the market. The main motive for entering ERP project was cost control and costing mechanisms.
2. **Medium sized chemical and pharmaceutical enterprise (p2);**
   The enterprise was crossing the edge between medium and large company. Many business processes were not streamlined and management wanted to reduce chaotic reactions to market disruptions by using strict logic of the ERP system. The decision to choose standard ERP solution was mostly based on the desire to reach competition at every possible level as soon as possible.
3. **Big glass industry enterprise (p3);**
   The huge international consortium took over the enterprise and wanted to impose strict cost control, detailed and documented conduction of business processes and production planning integration with other member plants. The decision to use standard ERP system was imposed by the new owner and the fact that ERP system was consortium standard.
4. **Radio and video broadcasting enterprise (s1);**
   The enterprise was fully owned by the government and was preparing for the complete deregulation of the media and broadcasting market. By the decision of the national parliament the enterprise was separated from the

national television to offer video and audio signal broadcasting services on the market. Since it was highly dependent on the legacy information system, which was not any longer in its possession but was left to the national television, the enterprise was desperately trying to establish its own information system as soon as possible. Therefore it chose standard ERP solution, which seemed fast and promising solution for given price-performance constraints.

5. **National railway company – transportation of passengers (s2);**
   The company was fully owned by the government and was part of the national railways. It was preparing for deregulation to become private owned, profit oriented enterprise for fast and convenient transportation of passengers. The management decided that one of the steps on this way was implementation of new information system. They found standard commercial ERP system to be most promising solution.
6. **National railway company – cargo transportation (s3);**
   The company was fully owned by the government and was part of the national railways. It was preparing for deregulation and tried to cut down the costs and modernize itself to become serious competition to other means of cargo transportation. The management considered information system as very important tool to track and manage costs. Their choice was standard ERP system as fastest possible solution.
7. **National railway company – railroad infrastructure (s4);**
   The company was fully owned by the government and was part of the national railways. In deregulated market it was still going to be owned by the government and would lease railroad infrastructure capacities to passenger and cargo transportation. Information system would be used primarily to support costs and planning of maintenance and development projects.
8. **National employment services (g1);**
   The services operated in circumstances of high unemployment rate and constant pressure from various stakeholders. The management decided to modernize services and make it capable to face rising competition on job market. For that step it needed information system to consolidate data and monitor overall performance. On the course of the project several ERP systems were analyzed as possible support for business processes, but final results showed that new information system should be custom designed and management accepted proposal.
9. **National ministry of labor and social welfare (g2);**
   The ministry operated in circumstances of high unemployment rate and tensed social relationships. It wanted to control high costs of its operations and rationalize social benefits system that was extremely expensive. To achieve these goals it urgently needed information system. During project several ERP systems were analyzed, but project strongly exposed arguments against standard off the shelf solution, which was accepted by the ministry.

All analyzed organizations intended to implement standard ERP solution or were in the process of decision whether to use standard or custom developed solution

Unfortunately in certain cases, decision for choice of the standard solution, was not based on deep business analysis and strategic planning, but mostly on desire to establish certain functionality of information system as soon as possible. Since such software solutions are usually marketed as support to best business practices, management also intended to improve business processes by changing them according to the requirements of software application. In these circumstances real strategic planning of information system was not possible. However, methods and techniques for strategic planning of information system, systemized in SPIS methodology, were applied in order to determine the real impact of standard solution on organization and its performance. Second goal of the described projects was to create architecture of complete information system composed of standard solution and custom developed modules, which should cover the whole organization and all the corresponding business processes.

In each of the projects, all relevant steps of SPIS methodology were applied. Methodology was modified to consider the fact that information system was not going to be completely developed but composed of standard and customized modules. Special emphasis was put on business process reengineering, and very significant work practice changes were proposed to maximize the influence of information system on business performance. Also the relationship was established between SPIS steps end ERP system implementation methods. Particular attention was directed towards gap analysis. Gap analysis revealed areas that were not supported by standard functionality of the ERP system. When related with other methods of strategic planning it could reveal complete new picture of the ERP system usefulness.

This paper describes relationship between gap and BCG analysis in abovementioned projects. BCG analysis groups business processes according to their strategic relevance and dependence on information technology. Combined with gap analysis, BCG matrix gives answer which business processes are supported by standard ERP modules and what is their strategic relevance. All organizations were thoroughly analyzed and business processes were classified according to the BCG criteria. The percentage of supported processes was calculated for each of the four BCG classes. Obtained results made possible comparison among analyzed organizations and sectors to which they belong.

Average duration of SPIS phase in abovementioned projects was approximately a year. The number of projects, in which results were collected, is not big enough to be statistically significant, but it certainly gives good insight to the fact that standard ERP systems quite often fail to support strategic and prospective business processes.

5 Results and discussion

The results of BCG analysis and gap analysis are presented in Table I and graphically in Figure 2. The table presents the percentage of all business processes in each of analyzed organizations supported by the standard functionality of an ERP system. The business processes are classified according to BCG criteria and organizations are grouped according to business sector to which they belong. Each analyzed organization has certain number of business processes in every quadrant of the BCG matrix and their classification was iterated several times and verified by the top management. During the gap analysis each process was checked against standard ERP functionality and the level of support was validated. Therefore in every quadrant of the BCG matrix two groups of business processes were obtained: supported and unsupported. The ratio of supported across overall number of business processes was calculated and is presented in Table 2.

The results of the research show that various groups of organizations are differently supported by standard functionality of the ERP systems. The best results are obtained in production industry. This is understandable because ERP systems have their origins in manufacturing systems. Also, variations between different business manufacturing systems are much less significant than in services oriented industry or public administration, so standard templates, on which ERP systems are based, fit relatively well. Differences between process oriented and discrete manufacturing systems might be quite significant but they are both well supported by most of the modern standard ERP packets. However, even in production domain, there is still much room for improvement and use of modern technologies in a manner that brings strategic advantage. The biggest support is on the level of supporting and operational business processes. These are usually financial, human resources, and standard production business processes. In production oriented organizational systems, strategic business processes are related to research and development, special features of production (e.g. integration of production equipment and information system) and warehousing. Prospective processes mainly deal with distribution of finished goods, operational maintenance and special features of supply chain management.

In services oriented organizations, standard ERP functionality gives much lover level of support. Again, better results are obtained at supportive and operational level, while strategic and prospective business processes are extremely poorly supported.

Public and governmental administration has low level of support at all four levels when compared to conventional production and services sector. This is quite understandable because their specifics are quite unique, especially for strategic and prospective level. Strategic and prospective processes of these organizations are related to improvement of their core business, which can hardly be found in standard ERP functionality. These are services to citizens (G2C), better communication with other governmental bodies (G2G) and various management issues that heavily relate on data analyses.

The results show that various sectors are not equally supported by standard ERP solutions and that in each sector level of support deteriorates towards strategic business processes. Without support for this kind of processes standard ERP system cannot bring strategic advantage to the organization. This is also true for prospective processes because they will never reach their full potential without proper use of ICT.

Table 2. Percentage of support that ERP system delivers to BCG classified business processes

	Production				Services					Public administration		
	p1	*p2*	*p3*	Avg.	*s1*	*s2*	*s3*	*s4*	Avg.	*g1*	*g2*	**Avg.**
Supporting	20	19	18.12	19.04	5.55	17.24	21.21	20	16	17.64	8.69	13.17
Operation	32	29.02	21.21	27.41	19.44	17.24	27.27	22.85	21.7	11.76	10.87	11.32
Strategic	17.14	16.03	14.13	15.77	13.88	0	12.12	8.57	8.64	0	6.52	3.26
Prospective	5.71	4.2	6.14	5.35	13.88	17.24	9.09	5.71	11.48	0	4.35	3.26

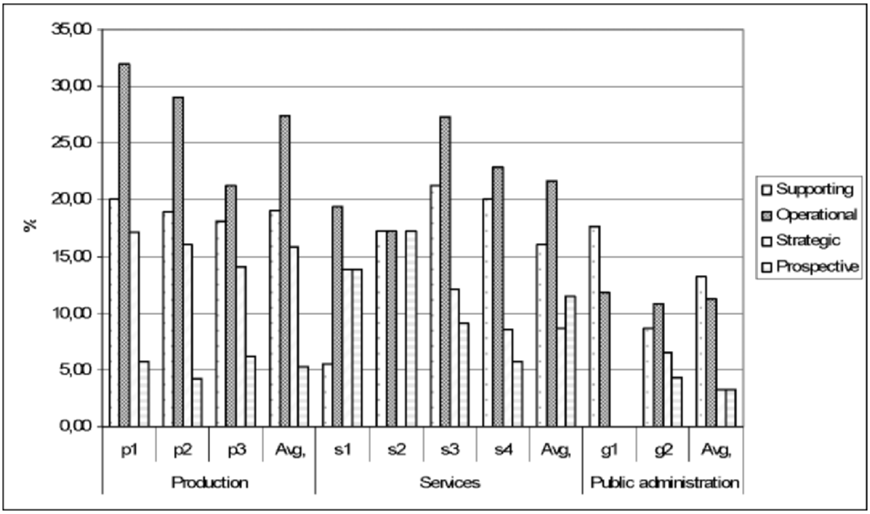


Fig. 2. Support of ERP system to BCG classes of business processes

6 Conclusion

Implementation of ERP system presents tremendous effort for the enterprise. Therefore it is normal that management, and organization as the whole, have high expectations from such modern software solutions. However long lasting implementation might lead to a disappointment if these expectations are not met. It might happen even in situations when implementation procedure was satisfying and all software modules were put in place. The reason for that comes from the fact that the key business processes were not supported by the standard software system. From the formal side such projects cannot be regarded as failures, but they simply do not bring competitive edge and adequate return of investment. They simply missed most important business processes.

Standard ERP system implementation methods do not pay enough attention to business process reengineering. They are too narrowly focused towards software implementation and deliberately avoid any broader context. Without severe rethinking of business work practice, significant changes in business processes, and detailed extension to integral information system such efforts are far from optimal.

Detailed business system analysis combined with strategic planning as in SPIS methodology gives comprehensive architecture of business processes and the architecture of information system, which should support them. Classification of these business processes with BCG matrix gives perspective of areas, which should be primarily supported to justify effort and investment. Special emphasis should be given to strategic business processes. These are processes, which give competitive advantage to the enterprise and differentiate it from the competition. Support for these processes cannot be bought off the shelf. ERP system must be implemented with clear vision of extensions that are going to be added to give new momentum to the enterprise.

Very important aspect is industry sector which is targeted by the implementation. Unfortunately ERP vendors enter all sectors promoting their software as silver bullet for all kinds of organizations. This is far from true. Results presented in this paper show that in government sector ERP system perform extremely poorly. However it is quite often that administration heavily invests in information technology and unfortunately frequently select ERP systems as solutions.

Therefore it can be concluded that ERP systems, although very expensive and complex, do not bring competitive advantage within boundaries of their standard functionality. Most often they are infrastructure on which specialized and strictly focused modules must be built. These modules will improve competitive edge and ERP system will give strength of integration, which will eventually lead to new business accomplishments or better services.

References

1. Abdinnour-Helm S, Lengnick-Hall ML, Lengnick-Hall CA (2003) Pre-implementation attitudes and organizational readiness for implementing an enterprise resource planning system. European Journal of Operational Research 146: 258–273
2. Al-Mashari M, Zairi M (2000) The effective application of SAP R/3: A proposed model of best practice. Logistics Information Management 13 (3): 56–166
3. Alpar P, Kim M (1990) A microeconomic approach to the measurement of information technology value. Journal of Management Information Systems 7(2): 55–69

4. Apperlrath H, Ritter J (2000) SAP R/3 Implementation: Method and Tools. Springer, Berlin
5. August-Wilhelm S, Habermann F (2000) Making ERP a success. Communications of the ACM 43(4): 57–61
6. Bancroft NH, Seip H, Sprengel A (1998) Implementing SAP R/3, 2nd edition, Manning Publications, Greenwich, CT
7. Bingi P, Sharma M, Godla J (1999) Critical issues affecting ERP implementation. Information Systems Management, 16(3): 7–14
8. Bradford M, Florin J (2003) Examining the role of innovation diffusion factors on the implementation success of enterprise resource planning systems. International Journal of Accounting Information Systems, 4: 205–225
9. Brumec J (1998) Strategic Planning of Information Systems. Journal of Information and Organizational Sciences, Vol. 23: 11–26
10. Brumec J, Dušak V, Vrček N (2001) The Strategic Approach to ERP System Design and Implementation. Journal of Information and Organizational Sciences, Vol. 24
11. Brumec J, Vrček N, Dušak V (2001) Framework for Strategic Planning of Information Systems. Proceedings of the 7th Americas Conference on Information Systems, Boston, MA, USA: 123–130
12. Brumec J, Vrček N (2002) Genetic Taxonomy: the Theoretical Source for IS Modelling Methods. Proceeding of ISRM2002 Conference, Las Vegas, NV, USA: 245–252
13. Brynjolfsson E, Hitt LM (1998) Beyond the productivity paradox. Communications of the ACM, 41(8): 49–55
14. Brynjolfsson E, Yang S (1996) Information technology and productivity: a review of the literature. Advanced Computing: 179–214
15. Brynjolfsson E (1993) The productivity paradox of information technology. Communications of the ACM, 26(12): 67–77
16. Chew W, Leonard-Barton D, Bohn R (1991) Beating Murphy’s Law. Sloan Management Review: 5–16
17. Cliffe S (1999) ERP implementation. Harvard Business Review 77 (1): 16–17
18. Davenport TH (1998) Putting the enterprise into the enterprise system. Harvard Business Review 76(4): 121–132
19. Davenport TH (2000) Mission Critical: Realizing the Promise of Enterprise Systems. Harvard Business School Press, Boston: 56–62
20. Davis B, Wilder C (1998) False starts, strong finishes—companies are saving troubled IT projects by admitting their mistakes, stepping back, scaling back, and moving on, Information Week 30: 41–43
21. Diederich T (1998) Bankrupt firm blames SAP for failure. Computer World, August 28: 35-38
22. Drucker PF (1998) The coming of the new organization. Harvard Business Review, 66(1): 45–53
23. Everdingen Y, Hillegersberg J, Waarts E (2000) ERP adoption by European midsize companies. Communications of the ACM, 43(4): 27–31

24. Gefen D (2002) Nurturing clients' trust to encourage engagement success during the customization of ERP systems. Omega 30: 287–299, http://www.elsevier.com/locate/dsw
25. Grady MW (1991) Performance measurement: Implementing strategy, Management Accounting (June): 49–53
26. Grover V, Teng J, Segars AH, Fielder K (1998) The influence of information technology diffusion and business process change on perceived productivity: the IS executive's perspective. Information Management 34(3): 141–59
27. Gurbaxani V, Whang S (1991) The impact of information systems on organizations and markets. Communications of ACM 34(1): 59–73
28. Hayes DC, Hunton JE, Reck JL (2001) Market reaction to ERP implementation announcements. Journal of Information Systems, 15(1): 3–18
29. Hitt LM, Brynjolfsson E (1996) Productivity, business profitability, and consumer surplus: three different measures of information technology value. MIS Quarterly, 20(2): 121–42
30. Hong KK, Kim YG (2002) The critical success factors for ERP implementation: an organizational fit perspective. Information & Management 40: 25–40
31. Hunton JE, McEwen RA, Wier B (2003) Analysts' reactions to ERP announcements., Journal of Information Systems: 57–64
32. Hunton JE, Lippincott B, Reck J (2003) Enterprise resource planning systems: comparing firm performance of adopters and nonadopters. International Journal of Accounting Information Systems, 4: 165–184
33. Kaplan R, Norton D (1992) The Balanced Scorecard – Measures That Drive Performance. Harvard Business Rewiev, Vol. 70., No 1.: 71–80
34. Kayes J (2005) Implementing the IT Balanced Scorecard. Auerbach Publications, Bocca Raton, FL, USA
35. Langdoc S (1998) ERP reality check for scared CIOs. PC Week 15 (38): 88
36. Laughlin S (1999) An ERP game plan, Journal of Business Strategy, (January–February): 32–37
37. Lynn M, Madison R (2000) The role of ERP software revisited. Journal of Accountancy 190(1), (July): 104–105
38. Mahmood MA, Mann GJ (1993) Measuring the organizational impact of information technology investment: an exploratory study. Journal of Management Information Systems, 10(1): 97–122
39. McFarlan FW (1984) Information technology changes the way you compete. HBR, March/April
40. McFarlan FW, Cash JI (1989) Competing through Information Technology. Harvard Business School Press., Boston, USA
41. Mitra S, Chaya AK (1996) Analyzing cost-effectiveness of organizations: the impact of information technology spending. Journal of Management Information Systems, 13(2): 29–57

42. Motwani J, Mirchandani D, Madan M, Gunasekaran A (2002) Successful implementation of ERP projects: evidence from two case studies. International Journal of Production Economy, 75(1–2): 83–96
43. Peffers K, Dos Santos BL (1996) Performance effects of innovative IT applications over time. IEEE Transactions on Engineering Management, 43(4): pp. 381–92
44. Piturro M (1999) How midsize companies are buying ERP. Journal of Accountancy 188(3): 41–48
45. Poston R, Grabski S (2001) Financial impacts of enterprise resource planning implementations. International Journal of Accounting Information Systems, (2): 271–294
46. Ragowshy A, Ahituv N, Neumann S (1996) Identifying the value and importance of an information system. Information Management, 31: 89–102
47. Scheer A, Habermann F (2000) Making ERP a success. Communications of the ACM 43 (4): 57–61
48. Stein T (1997) Andersen consulting sued over R/3 installation. Information Week, (654), (October 27): 34
49. Umble EJ, Haft RR, Umble M (2003) Enterprise resource planning: Implementation procedures and critical success factors. European Journal of Operational Research, 146: pp. 241–257
50. Vincent A, Mabert L, Ashok Soni MA (2003) Enterprise resource planning: Managing the implementation process. European Journal of Operational Research 146: 302–314

Approach to Enterprise Knowledge Base Development

Saulius Gudas[*,**], Rasa Brundzaite[*]

[*]Vilnius University, Kaunas Faculty of Humanities, Muitines 8, LT-44280 Kaunas, Lithuania, {gudas, rasa.brundzaite}@vukhf.lt
[**]Kaunas University of Technology, Information Systems Department, Studentu 50, LT-51368 Kaunas, Lithuania, gudas@soften.ktu.lt

Abstract: Approach to Enterprise Knowledge Base development is based on the redefined concept of the knowledge-based enterprise and follows the Knowledge-based IS engineering paradigm. The Enterprise Knowledge component (Business, IT, Knowledge) is the core knowledge item of the presented approach. Enterprise Knowledge Space is formed for the analysis of the semantics and granularity of enterprise knowledge items. The UML class diagram of the Enterprise Knowledge Base is presented as a step towards practical development for the Enterprise Knowledge Base.

1 Introduction

Contemporary highly networked organizations use information technology (IT) extensively and need for an organizational memory accessible in the virtual environment. Well structured, validated and shared computerized part of the organizational memory is termed here as the Enterprise Knowledge Base (KB) and is intended to support management decision making, business and IT alignment as well as knowledge management (KM) and information systems (IS) development processes.

The knowledge-based theory of the firm considers knowledge as the most strategically significant resource of the firm [1, 2]. Information technologies play an important role in the knowledge-based view of the firm in sense that information systems can be used to synthesize, enhance, and expedite large scale intra- and inter-firm knowledge management [3]. There are a variety of KM models [4] and tools, but firms still face challenges when implementing IT systems for the support of KM [5]. Organizations require more systematic and formalized methods for the IT-based knowledge management implementation.

Enterprise KB is concerned as the obligatory component of the knowledge-based enterprise. The research starts with the identification of the core domains of the knowledge-based enterprise with the aim to clarify the role and the place of the Enterprise KB within the knowledge-based enterprise architecture.

In the chapter four the scope and boundaries of the enterprise knowledge are analyzed. In the chapter five the Enterprise Knowledge Space is defined with the aim to create practical method for Enterprise KB development. The basic components of the Enterprise KB are refined and represented by UML class diagram in the sixth chapter thus forming the background for the development of the Enterprise KB management system.

The presented models concern the knowledge management process in the knowledge-based enterprises. In the article the concept of the *knowledge-based enterprise* is used, which is based on the resource-based view of the firm [1,2], the conception of the *knowledge-based business*, provided by Zack [6] and follows the *knowledge-based information systems engineering* paradigm described by Gudas, Skersys, Lopata [7].

2 The Components of the Knowledge-Based Enterprise

The knowledge-based enterprise, regardless of whether its products are tangible or not, here is defined according the concept of knowledge-based organization, presented by M.H. Zack [6]; namely, the *knowledge-based enterprise*: a) recognizes knowledge as a key strategic resource, b) rethinks its business processes in the knowledge-oriented sense (i.e. "it takes knowledge into account in every aspect of its operation and treats every activity as a potentially knowledge-enhancing act." [6]), c) aligns its knowledge management activity with its strategy. Besides, in the article it is suggested, that knowledge-based enterprise uses Enterprise Knowledge Base (KB) together with explicitly modelled knowledge management activity as obligatory enterprise management and enterprise IS development component. This suggestion is derived from the knowledge-based IS engineering paradigm. Gudas, Lopata, Skersys [7] deal with the advanced IS engineering methods and tools and introduce the concept of the Enterprise Repository (Knowledge Base) as the main component for the intellectualization of the CASE tools as well as whole IS engineering process. The shared Enterprise Repository stores Enterprise Models and serves as the main knowledge source for the accomplishing IS engineering tasks as well as for automatic generation of IS models. Research, presented in this article, is directed toward extending the functionality of the Enterprise Repository for knowledge-based enterprise design, management and business-IT alignment tasks.

The concept of the knowledge-based enterprise is illustrated further by using Strategic Alignment Model (SAM) [8]. The content and role of Enterprise KB within the knowledge-based enterprise architecture is explored too.

In the knowledge management literature there are various KM definitions presented (listed by R. Maier [9]). Knowledge management can be considered as a strategic process of an enterprise and is intended to solve critical enterprise adaptability and competitiveness problems in the rapidly changing environment. The main goal of the knowledge management in enterprises is to create organizational context for effective creation, store, dissemination and use of organizational knowledge, which are essential for securing enterprise competitiveness against changing business environment and for setting environment towards a desirable direction. The newest vision of the most adaptive and responsive enterprise, based on IT, is expressed by Gartner Group [10] as the Real Time Enterprise (RTE). The RTE is the kind of the enterprise, which is able to gain the synergetic combination of IT, knowledge management and business strategy. Y. Malhotra [11] have analyzed the knowledge gaps which arise when implementing knowledge management in RTE and pointed out two main KM models: strategy-pull and technology-push, thus indicating two interrelated RTE domains: business (strategy) domain and technology domain. Henderson and Venkatraman [8] have also analysed business-IT alignment problem and proposed a Strategic Alignment Model (SAM) for business–IT alignment; the model was aimed to support the integration of information technology (IT) into business strategy by advocating alignment between and within four domains. In the SAM two interrelated aspects of computerized enterprise are defined: 1) Business domain and 2) IT domain; both domains are viewed in two levels of detail: 1) Infrastructure and processes level, 2) Strategic level. As a result of such decomposition four different domains were identified: Business strategy domain, Business infrastructure and processes domain, IT strategy domain and IT infrastructure and processes domain (see Fig.1).

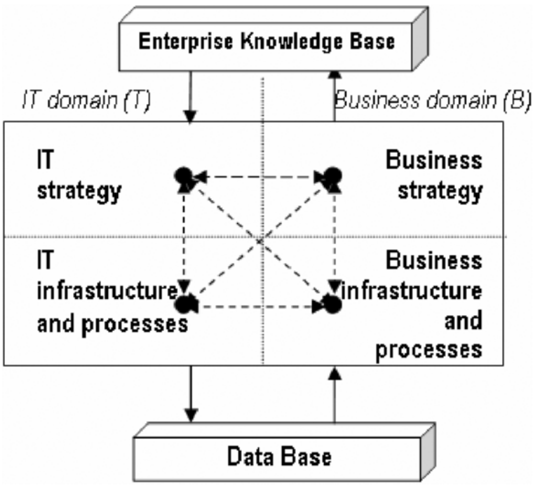


Fig. 1. Integration of enterprise components, related to knowledge management

Thus SAM model identifies major interactions between business and IT domain, which can be expressed as structural models.

Knowledge-centric enterprise, as any other contemporary organization, possibly uses the integrated data repositories which are presented in the SAM as the *Data Base* component (Fig.1).

In order to support alignment decision making between four domain, Strategic Alignment Model is complemented by one more additional structural element – *Enterprise Knowledge Base* element (see Fig.1), which supports enterprise knowledge management activities and allows continuous cross-domain alignment process for gaining the synergetic combination of both (business and IT) domains.

The scope and boundaries of the enterprise knowledge will be examined in the next chapter. First let us discuss the types of knowledge the Enterprise KB comprises.

Organizational memory comprises all the possible forms of organizational knowledge: tacit, explicit, computerized, not-computerized etc. [9].

There are a lot of possible facets for characterising knowledge [9] although it is important in this situation to analyse knowledge in the sense of its “objective” and “subjective” characteristics. According J.M Firestone [12], there are two kinds of knowledge: “Knowledge viewed as belief…”, and “Knowledge viewed as validated models, theories, arguments, descriptions, problem statements etc.”

In the area of IS engineering Enterprise KB is the source of knowledge about the problem domain (i.e. enterprise). Business-IT alignment is continuous decision making process and it also should be supported with reliable information and knowledge. Resuming it should be stated that Enterprise Knowledge Base stores the enterprise knowledge in the form of validated Enterprise Knowledge Models for business management and computerization.

According knowledge-based IS engineering paradigm [7, 13] enterprise knowledge models have to be validated according formalized enterprise model thus ensuring reliability of the acquired knowledge about problem domain.

So, Enterprise Knowledge Base is the reliable knowledge source for a support of business management decision making, business and IT alignment as well as for support of knowledge management and IS development processes.

3 Four Domains of the Knowledge-Based Enterprise

In SAM [8] two interrelated aspects of the enterprise are defined: 1) business domain and 2) IT domain. According the definition of the knowledge-based enterprise, another two important aspects of the enterprise can be identified: Knowledge domain and Data domain (Fig.1). All these four aspects forms four interrelated domains (see Fig.2), which have to be taken into account when transforming business into knowledge-based business.

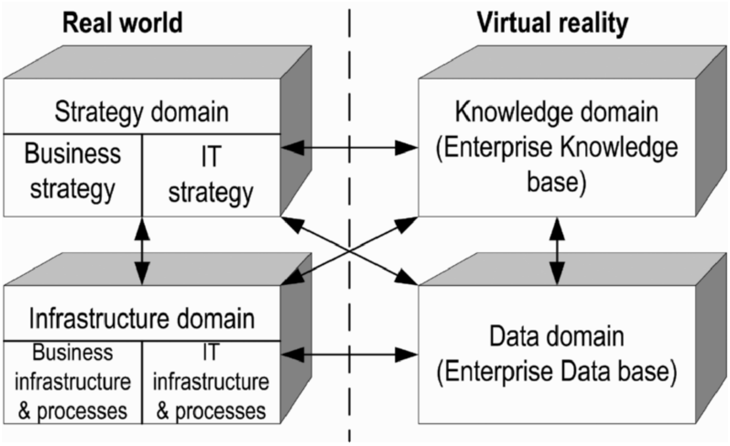


Fig. 2. Four domains of the knowledge-based enterprise

The peculiarity of the abstraction, presented in the Fig.2 is that it clearly separates the Knowledge domain and Data domain, in contrast to other conceptual enterprise models (e.g. presented in [14]).

Knowledge-based enterprise uses knowledge as a key strategic resource, as it was said before (see also Fig.2). It became evident that the *organizational knowledge (non-digital knowledge)* is human knowledge used (and hidden in the Fig. 2) in the *business* and *IT domains* as integral components of any enterprise.

Meanwhile *enterprise knowledge* (virtual, digital knowledge stored in the Enterprise KB) is obligatory component of knowledge-based enterprise, integrated with all enterprise domains. Accordingly three tiers Knowledge-based Enterprise Architecture (in the Fig.3), it includes Enterprise KB as key component, integrated with knowledge management systems.

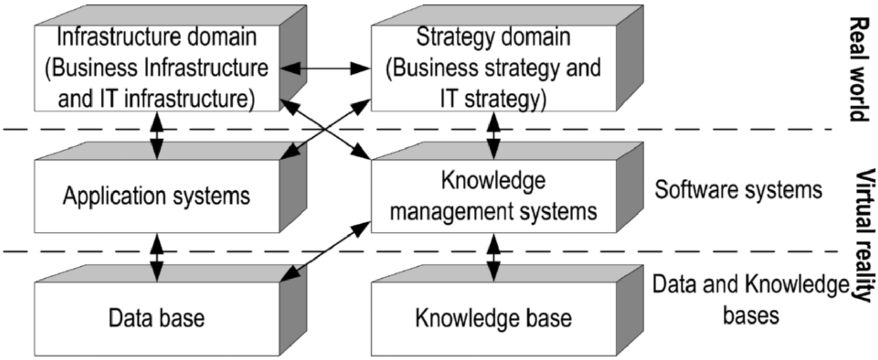


Fig. 3. Knowledge-based Enterprise Architecture

The following chapter is focused on the identification of the boundaries and actual content of the Enterprise Knowledge Base.

4 Modelling Views of the Enterprise Knowledge and Data

It should be pointed out two types of knowledge inherent to the knowledge-based enterprise.

First type of knowledge comprises all the organizational memory, which consists of various types of human knowledge, handled by managers daily to perform organizational activities as well as for management of these activities. This type of knowledge is referred to as *organizational knowledge.*

Another type of knowledge is a subset of knowledge stored in the Enterprise KB, and is named as *enterprise knowledge.* It comprises virtual (digital) knowledge about the problem domain, i.e. activities of knowledge-based enterprise.

From the abstraction shown in the Fig.2 it can be concluded that knowledge-based enterprise has four interrelated aspects: IT strategy and infrastructure (*T*), Business strategy and infrastructure (*B*), Enterprise knowledge (*K*) and Data. The major aspects of enterprise activities (*T, B, K*) are interrelated and comprises enterprise knowledge component (see Fig.4).

The enterprise knowledge component is associated with enterprise data repositories (data base) as well (see Fig.5).

The depicted (Figs.4,5) enterprise knowledge component structure represents a structural viewpoint to enterprise knowledge modelling: the Knowledge Base should include *integrated enterprise knowledge* (validated models, theories, arguments, descriptions, problem statements) about business strategy and infrastructure (*B*), IT strategy and infrastructure (*T*), and *enterprise modelling knowledge* (*K*).

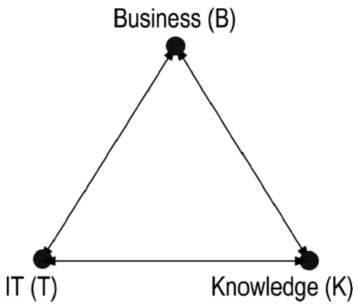


Fig. 4. Enterprise knowledge component

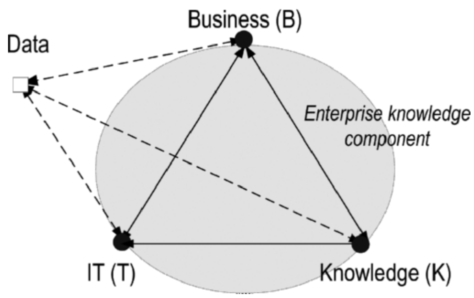


Fig. 5. Integration of enterprise knowledge and data

The detailed description of these three aspects of enterprise knowledge is the next chapter and illustrated in the Fig.6.

5 The Enterprise Knowledge Space

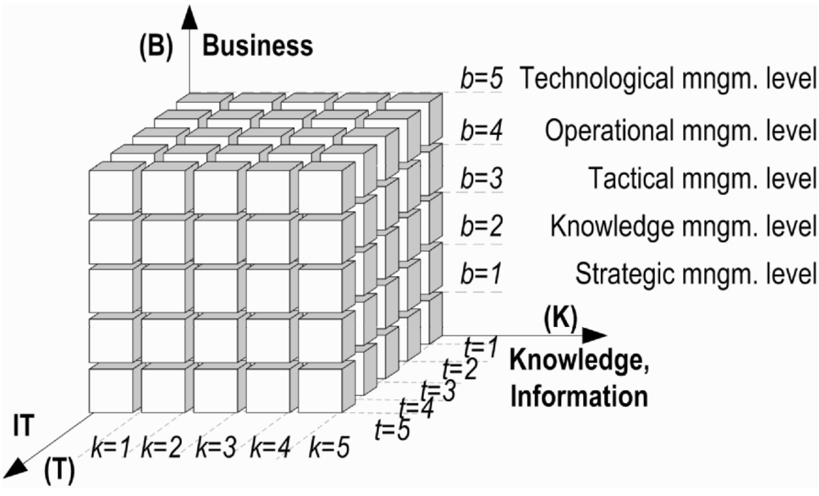
The next step is to identify the analysis levels of each enterprise knowledge aspect.

The Enterprise Knowledge Space *E (B, T, K),* presented in Fig.6, is constructed for systematization of the enterprise knowledge research with the aim to find practical enterprise knowledge modelling method.

The contemporary organizational theories distinguish between four hierarchical levels in organizations: strategic management level, tactical management level, knowledge management level and operational management level [15].

On the basis of such a hierarchical system, we can define four adequate levels in organizational information management processes as well as technological processes management level. We have modified a slightly ordinary hierarchical structure by placing knowledge management in the second level of the management hierarchy, because of the overall nature of the knowledge management processes.

The enterprise knowledge space (Fig.6) was formed by fitting integrated knowledge component with the hierarchical information structure of the organization as mentioned above.



:

Fig. 6. The Enterprise Knowledge Space *E(B,T,K)*

Each item *e* in the enterprise knowledge space *E(B,T,K)* is identified according to three axes:

$$e(b;t;k) \in E, \quad \forall b,t,k \in \{1,...,5\}. \tag{5.1}$$

Each of the 125 items has its own semantics and identifies the element of enterprise knowledge, which integrates three aspects of the enterprise: business (*B*), information technology (*IT*) and knowledge (*K*) in 5 levels of detail (accordingly *b,t,k*).

Each index *b, t* and *k* can acquire values from following sets:

$\forall b \in$ *{b1=Strategic management level; b2= Knowledge management level; b3= Tactical management level; b4= Operational management level; b5=Technological processes management level};* (5.2)

$\forall z \in$ *{z1=Ontological level; z2=Meta-metamodelling level; z3=Metamodelling level; z4=Conceptual modelling level; z5=Business (knowledge, applied in business) level}.* (5.3)

$\forall t \in$ *{t1=User interface level; t2= Enterprise data (structures) level; t3=Business task logic level; t4=Network level; t5= IT infrastructure level};* (5.4)

Knowledge axis (*K*) is divided into the 5 levels of detail (see Eq.4.3) according recommendations of the MDA [16]. And the third, IT axis is divided into five levels of detail (see Eq.4.4) on the ground of existing IT architecture principles.

The Enterprise Knowledge Space can be treated as a morphological box, which models various interactions between those three integrated enterprise aspects: business (*B*), information technology (*IT*) and organizational knowledge (*K*).

It is possible to abstract three two-dimensional spaces in the enterprise knowledge space: (*B*x*T*), (*B*x*K*), (*T*x*K*). Each of the two-dimensional spaces simulates the interactions of each two aspects from the Enterprise Knowledge Space. These two-dimensional models logically interrelate with such well-known models like Information System Architecture model by J.Zachman [17] and Multi-perspective enterprise modelling [18]. However, these models are quite different and the research of these two-dimensional models is outside the boundaries of our research.

6 The Major Classes of Enterprise Knowledge Base

The Enterprise Knowledge Model *M* is derived from formal description of Enterprise Knowledge Space.

The formal description of the Knowledge-based Enterprise Knowledge Model *M* can be expressed as the Cartesian product in the following way:

$$M = (T) \times (K) \times (B) \times (R) \tag{6.1}$$

where T = information technology, K = knowledge, B = business process, R = business resources.

For the completeness of the model, *resources* (*R*) component was introduced into the model, as we consider knowledge as separate, but integrated enterprise aspect in contrary to the classical enterprise modelling methods which analyze knowledge alongside with other business resources.

This means, that each enterprise knowledge item *m* ($m \in M$) in Enterprise Knowledge Model *M* is related to the appropriate business process *b* ($b \in B$), knowledge *k* ($k \in K$), resources *r* ($r \in R$) and information technology *t* ($t \in T$). To put it in other terms, the enterprise knowledge model *M* item *m* is identified by a set of identifiers:

$$m(t;k;b;r;l) \in M \tag{6.2}$$

where $l\ (l \in L)$ is time period index.

The Enterprise Knowledge Model *M* is composed of interrelated items *m* and enables modelling a knowledge-based business, as it considers enterprise modelling knowledge (identifier *k*), business processes (identifier *b*), IT (identifier *t*), and their interactions. Each aspect (*b,t,k*) is modelled into the 5 levels of detail.

The enterprise knowledge modelling method and tool for practical business management needs can be developed on the basis of the Enterprise Knowledge Model.

In Fig.7 the class diagram (UML Class model) of the Enterprise Knowledge Model is presented.

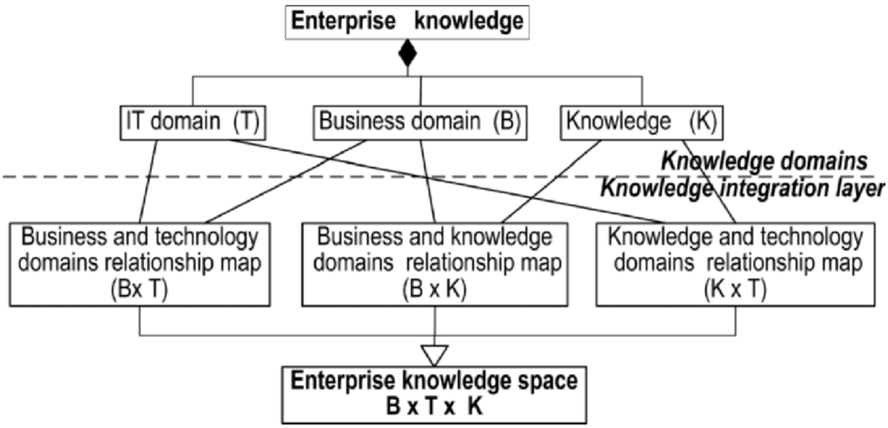


Fig. 7. Enterprise knowledge structure (subsets of knowledge), presented as UML class diagram

Consequently, Enterprise Knowledge Base contains integrated knowledge about three enterprise domains: business (*B*), information technology (*T*) and knowledge (*K*) as well as various relationships of these domains (represented as knowledge integration layer in Fig.7).

7 Conclusions

Several conceptual findings concerning knowledge management in the enterprise were made thus making background for the development the practical methods for the systematic implementation of the knowledge management elements in the enterprises and for the transformation into the knowledge-based enterprise.

Redefined concept of the knowledge-based enterprise led to the identification of its main architectural elements. The Enterprise Knowledge Base is concerned as main component of the knowledge based enterprise. The role and scope of the Enterprise Knowledge Base (KB) were specified as well as the interactions with other architectural elements.

The structuring of the knowledge-based enterprise according to the Strategic Alignment Model by Henderson and Venkatraman enabled the identification of four domains of the knowledge-based enterprise. According such as division of the knowledge-based enterprise, three components of the enterprise knowledge component were abstracted: business (B), enterprise knowledge (K) and information technologies (T).

For the explanation of the semantics of the enterprise knowledge according all three aspects, the Enterprise Knowledge Space was developed.

Formal description of the Enterprise Knowledge Space is assumed as the Enterprise Knowledge Model M, which forms background for the creation of practical method for the Enterprise KB development.

Finally, on the basis of the developed model Enterprise Knowledge Model M, the UML class diagram of the Enterprise Knowledge Base is presented as a step towards practical development for the Enterprise KB.

Enterprise KB integrates enterprise knowledge about three interrelated domains: business (B), enterprise knowledge (K) and information technologies (T), thus enabling support of the knowledge-based enterprise design as well as its IS engineering.

It is expected that the introduction of integrated enterprise knowledge base into the overall business management and development framework will improve on the enterprise's knowledge management, adaptability and flexibility.

References

1. Grant RM (1996) Toward a knowledge-based theory of the firm. In: Strategic Management J. 17, pp 109–122
2. Conner KR (1991) A historical comparison of resource-based theory and Five schools of thought within industrial organization economics: Do we have a new theory of the firm? In: J. Management 17(1), pp 121–154
3. Alavi M, Leidner, DE (2001) Review: Knowledge Management and Knowledge Management Systems, MIS Quarterly (25:1), March 2001, pp 107-136

4. Holsapple CW, Joshi KD (1999) Description and Analysis of Existing Knowledge Management Frameworks. In: Proceedings oh the 32nd Hawaii International Conference on System Sciences, 1999 vol. 1, pp 1072-1087, available at: http://csdl.computer.org/comp/proceedings/hicss/1999/0001/01/00011073.PDF
5. Malhotra Y.(2004) Why knowledge management systems fail. Enablers and constraints of knowledge management in human enterprises. Information Today Inc., Medford, NJ, pp 87-112, available at: www.brint.org/WhyKMSFail.htm
6. Zack MH (2003) Rethinking the knowledge-based organization. *Sloan management review*, vol. 44, no. 4, summer 2003, pp 67-71, available at: http://web.cba.neu.edu/~mzack/articles/kbo/kbo.htm
7. Gudas S, Skersys T, Lopata A (2004) Framework for knowledge-based IS engineering. In: Advances in Information Systems ADVIS'2004, T. Yakhno, Ed., Berlin: Springer-Verlag, 2004, pp 512-522
8. Henderson J, Venkatraman N (1990) Strategic alignment: A model for organization transformation via information technology. Massachusetts Institute of Technology, Working Paper 3223-90
9. Maier R (2004) Knowledge Management Systems: Information and Communication Technologies for Knowledge Management. Springer; 2nd ed
10. Now is the Time for Real-Time Enterprise (2002) Gartner Group, available at: http://www.gartner.com/pages/story.php.id.2632.s.8.jsp
11. Malhotra Y (2005) Integrating knowledge management technologies in organizational business processes: getting real time enterprises to deliver real business performance. In: Journal of Knowledge Management, vol. 9, no. 1, pp 7-28
12. Firestone JM (2001) Knowledge Management: A Framework for Analysis and Measurement. In: DsStar, vol.5, No.1, available at: http://www.taborcommunications.com/dsstar/01/0102/102505.html
13. Gudas S (2005) Žiniomis grindžiamos IS inžinerijos metodų principai [The principles of the Knowledge-based IS Engineering methods], In: IT'2006, pp 713-717
14. Iyer B, Gottlieb R (2004) The Four-Domain Architecture: An approach to support enterprise architecture design. IBM Systems Journal, vol. 43, no. 3
15. Laudon JP, Laudon KC (2002) Management Information Systems. Managing the Digital Firm (7th ed.). Prentice Hall
16. Hettinger MK (2003) Model-Driven Architecture, processes and methodology from the perspective of the "Modeling Discipline", In: MDA™ Implementers' Workshop Succeeding with Model Driven Systems
17. Zachman JA, Sowa JF (1992) Extending and Formalizing the Framework for Information Systems Architecture. In: IBM Systems Journal, vol. 31, no. 3

18. Ulrich F (2002) Multi-Perspective Enterprise Modeling (MEMO) – Conceptual Framework and Modeling Languages. In: 35-th Hawaii International Conference on System Sciences – 2002, available at: csdl.computer.org/comp/proceedings/hicss/2002/1435/03/14350072.pdf

Co-designing Models for Enterprises and Information Systems – A Case for Language Integration

Peter Rittgen

University College of Borås, 501 90 Borås, Sweden, peter.rittgen@hb.se

Abstract: To achieve a close alignment of information systems and enterprises their designs have to be interwoven to a mutually supportive pattern. This requires compatible languages for expressing the designs. We suggest a framework for integrating two hitherto distinct languages specialized for the respective domain, UML and SIMM. With the help of a case study we demonstrate that this integration does indeed support the co-design of an enterprise model and an information systems model.

1 Introduction

The term co-design can be understood in different ways. In the context of our study it refers to an idea that was elaborated in (Forsgren 2005). It has its roots in "systems thinking" as established by (Churchman 1968). His principal idea was that we can design an unlimited number of views on reality. They may differ in their granularity (level of detail), their perspective, their level of abstraction, and so on. But from Churchman's point of view this is not sufficient. We must also "calibrate" the viewing instrument (or measurement scale) to arrive at (or agree on) a view that is supposed to be "implemented". This collective process of designing views and moving towards a consensus view is called co-design. It has shaped the way we look at social systems in general and information systems in particular (Ackoff 1981; Checkland 1988; Mitroff and Mason 1981). But what are the implications of this idea when the objects of our design are models? We will return to this point later.

When we take a systematic approach to co-design we might ask what the co in co-design refers to. The most obvious answer is perhaps that a number of designers jointly develop a design. We speak in that case of the subjects **in** co-design. But the design process also has a number of customers, the ones we design for.

We call them the subjects **of** co-design. If we further broaden our view we might also discover that design is about objects, and again we can identify two dimensions, the objects **in** and **of** co-design. The objects **of** co-design are the things that we design, the results of the design process. But in the process we also need tools or artefacts, the objects **in** co-design.

Our study is primarily about the latter category, i.e. the tools or artefacts. In modeling, the primary tools are languages. But languages have a problematic double role in being both objects **in** and **of** co-design. We use a language to describe, for example an enterprise or an information system. In that sense the language is a tool **in** design. But at the same time, when using the language we discover it new, reinvent it and put it to a different use by reinterpreting existing concepts of the language and creating new ones. Each use situation therefore leads to a change of the language. In that sense the language is also an object **of** co-design.

If languages are shaped by their use, then different contexts of use will also yield different languages. An enterprise modeling language will turn out to be different from a modeling language for information systems because they are used for describing different things. But although an enterprise and an information system are not the same thing, they do share an intimate relationship: The information system (of an enterprise) is a subsystem of that enterprise.

Let us summarize these important points. Languages are both design tools and design objects. As tools for modeling enterprises and information systems they are used on two objects **of** co-design where one is a part of the other. This situation makes the co-design of enterprises and their information systems an intricate business. The separation of enterprise modeling and information systems modeling into different areas of concern has led to the development of completely different languages for the respective areas (see section 2). But a co-design of enterprises and information systems is only possible when the modeling languages for these areas are also co-designed. Both the current situation and the future scenario of co-designed languages are depicted in fig. 1.

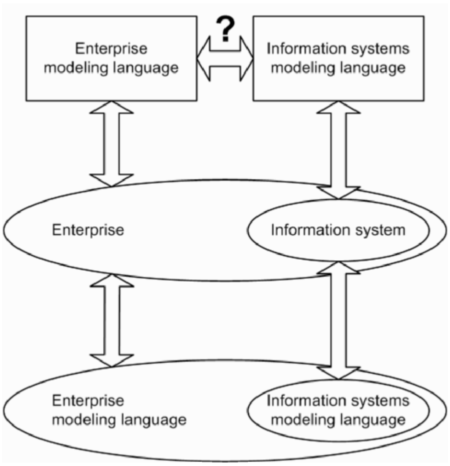


Fig. 1. Co-design of systems and languages

But how do we get started on such a process of co-designing languages? One approach would be to start at zero with “empty” languages, building both languages gradually from scratch in a co-design context, i.e. in projects that apply co-design thinking to develop an enterprise together with its information systems. But much of the time in this process would be spent on discovering concepts that already exist in current languages for enterprise modeling and information systems modeling. We therefore suggest another approach that avoids reinventing the wheel. We propose to investigate existing languages and to identify those concepts that could provide a “common ground” for language co-design. We call this process language integration. It consists of an analysis of the respective languages and their comparison (see section 2). Based on this we derive a framework for integration (see section 3) which we subsequently use for the co-design of business processes and information systems in the context of a case study (see section 4).

2 Language Analysis and Comparison

As candidate languages for our study we have chosen SIMM and UML. UML seemed a natural choice due to its high degree of standardization and its dominance in information systems modeling. Enterprise modeling, on the other hand, is less standardized and our choice of SIMM is hence less obvious. The justification for this particular method lies in a combination of a strong theoretical foundation and a significant empirical support. SIMM has its roots in Business Action Theory (Goldkuhl 1996, 1998; Goldkuhl and Lind 2004) and has undergone considerable empirical validation (Axelsson et al. 2000; Axelsson and Segerkvist 2001; Goldkuhl and Melin 2001; Haraldson and Lind 2005; Johansson and Axelsson 2004, 2005; Lind and Goldkuhl 1997; Lind et al. 2003; Melin and Axelsson 2004; Melin and Goldkuhl 1999).

The analysis of SIMM is based on the language specification (Röstlinger and Goldkuhl 2005). As SIMM is based on the ontology of Socio-Instrumental Pragmatism (SIP) (Goldkuhl 2002; Goldkuhl 2005) we have also made use of this information. Our analysis of UML is based on the UML 2.0 Superstructure Specification (OMG 2004) and the Infrastructure document (OMG 2006). In addition we make use of the ontological analyses of UML done in (Evermann and Wand 2001) and (Opdahl and Henderson-Sellers 2002). They employ the ontology by Bunge-Wand-Weber (BWW) which is an established tool for analyzing modelling languages. It is based on Mario Bunge’s ontology (Bunge 1977, 1979) and was later adapted by Yair Wand and Ron Weber to the information systems field (Wand and Weber 1989, 1995; Weber 1997).

As an example of how language analysis and comparison are performed we look at three predominant concepts: Actors, actions and action objects.

2.1 Actors

In theories of social action an actor is always a human being. But SIP recognizes that there can be non-human agency. An artefact (e.g. a computing system) can perform actions and its actions can have a meaning in the social world. Artefacts can play the role of non-human agents, whereas actors are human agents. It is therefore that we speak of a “performer” in the context of SIMM, encompassing human and non-human performers.

In UML the concept of an actor is much broader and covers that of a performer in SIMM plus ‘time’ which is also an actor in UML (but not in SIMM). We might therefore map SIMM-performer to UML-actor. But in UML the concept of an actor is restricted to the particular context of use cases. Otherwise actions are considered to be performed by objects. This is due to the fact that the UML is primarily a design language for (software) artefacts. Another possible mapping is therefore the one from SIMM-performer to UML-object which allows for a proper translation of interaction graphs.

2.2 Actions

The concept of action exists in both SIMM and UML and the language descriptions agree largely. It is therefore valid to map SIMM-action to UML-action. It should be noted, though, that SIMM provides a more sophisticated concept of human action as purposeful, social action that is performed with the help of some instrument (artefact), whereas non-human action by artefacts is secondary. In UML these roles are reversed but there is no general conflict (only a shift in focus). SIMM also defines interaction as a special form of action directed towards another actor. Action objects involved in an interaction become interaction objects in that case (see 2.3). This has no effect on the nature of the objects themselves, though.

2.3 Action objects

The term action objects is an explicit concept of SIMM and it refers to objects that are involved in an action. As human action is purposeful it is performed to achieve some result(s). Action objects produced by an action can be such results. But action objects serve also as input for other actions. In the UML objects are a fundamental concept. They constitute a system and all behavior of that system consists basically of messages that are exchanged between objects and the objects’ internal behavior. An UML-object can be the resource for another object and thereby play the role of a SIMM-action object. The mapping from SIMM-action object to UML-object is therefore valid. But in addition to that an UML-object can also be the performer of some action as discussed in 2.1.

3 Deriving the Integration Framework

We have extended the comparison described in section 3 to the remaining concepts. For this purpose we made use of the language specification (OMG 2004, 2006; Röstlinger and Goldkuhl 2005) but also of the more detailed information available from the ontological analyses (Evermann and Wand 2001; Goldkuhl 2002; Goldkuhl 2005; Opdahl and Henderson-Sellers 2002). By following this process we arrived at suitable matches for all SIMM concepts. The resulting mappings are shown in fig. 2 in the form SIMM concept → UML concept.

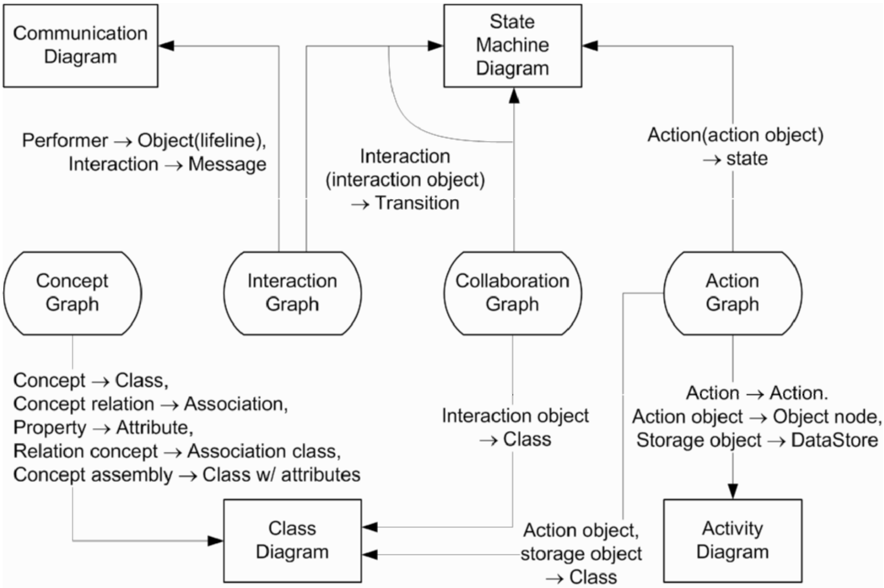


Fig. 2. The integration framework

But the mappings do not only help us in translating concepts from one language to another. The SIMM concepts are related to specific SIMM graphs and the UML concepts are related to certain UML diagrams. If we follow up these relations we can also establish a relation between SIMM graphs and UML diagrams, i.e. we can determine which information from which SIMM graphs we need to construct a certain UML diagram. In other words, we can derive a framework that supports the translation of SIMM models into UML models (see fig. 2).

To test the feasibility of this approach we used the framework in a project where we developed enterprise computing models in UML based on an enterprise model in SIMM. The following section reports on our experiences from that case study.

4 Co-designing Models: A Case Study

The case we investigated involved a retail chain in the home textile and home decoration industry. The logistics operation of that company had been outsourced to a third-party logistics provider. In the beginning of the project we performed an analysis of the business situation that led to an enterprise model that we documented with the help of the SIMM method. Two of the SIMM graphs serve as an example of this model, the interaction graph (fig. 3) and the action graph (fig. 4).

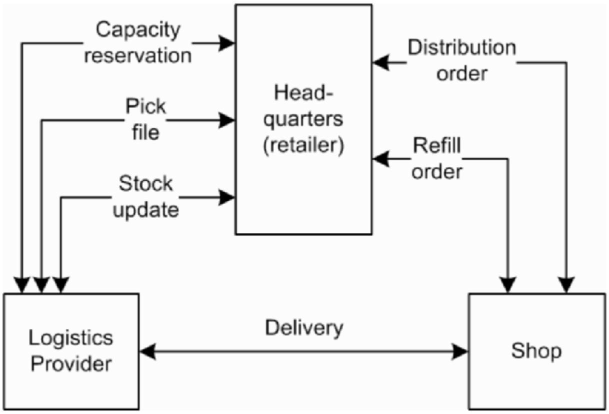


Fig. 3. Interaction graph

The main actors are the Logistics Provider (LogPro), the Headquarters of the retailer and a Shop. Fig. 3 shows the interactions between them with respect to order handling. It starts when Headquarters reserve capacity for handling a certain amount of items in advance of the actual order. LogPro allocates staff and space to provide for this capacity. The product assortment of the retailer consists of basic-range products and seasonal products. The latter are distributed to the Shop according to turnover quota (distribution order). This is triggered by Headquarters. Orders for basic-range products are initiated by the Shop. This happens when the Shop is running low on certain products (refill order). Headquarters forward both types of orders to LogPro in form of a pick list. LogPro performs delivery to the Shop. The confirmation can be accompanied by a complaint if items are missing or wrong ones have been sent. Periodically Headquarters ask for a stock update. This is necessary because they run their own warehouse management system which is not integrated with that of LogPro.

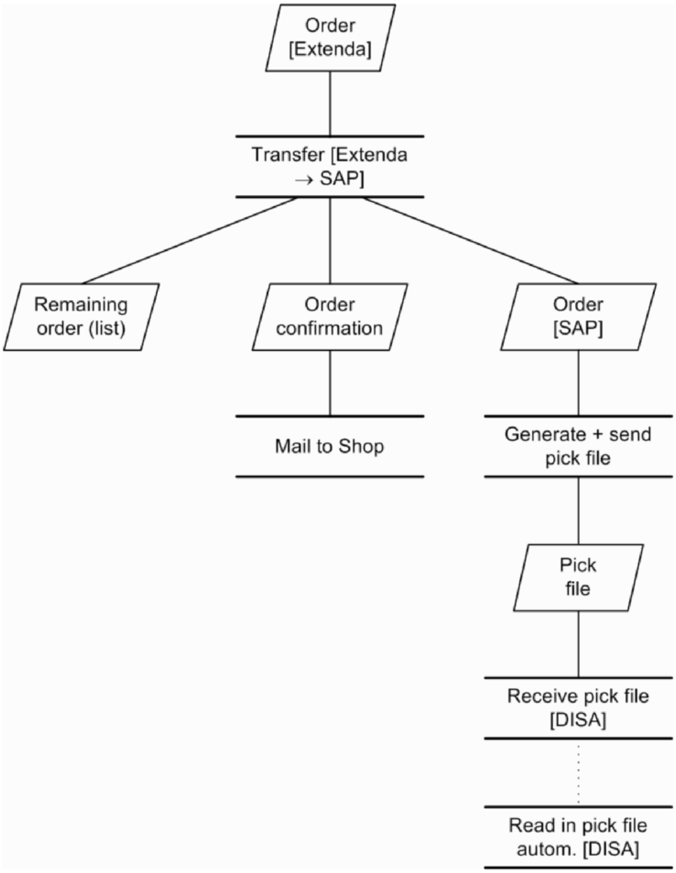


Fig. 4. Action graph

We take a closer look at one aspect of order processing, i.e. processing of the pick file. This routine requires the integration of different computing systems, the Extenda system of the Shop, the SAP system of Headquarters and the DISA system of LogPro. Fig. 4 shows an excerpt of the action graph for pick file processing. Based on the enterprise models we developed the UML models to support the design of respective enterprise computing systems. To give the reader an idea of this process we show the communication diagram (fig. 5) and the activity diagram (fig. 6) that have been derived from the interaction graph (fig. 3) and the action graph (fig. 4), respectively.

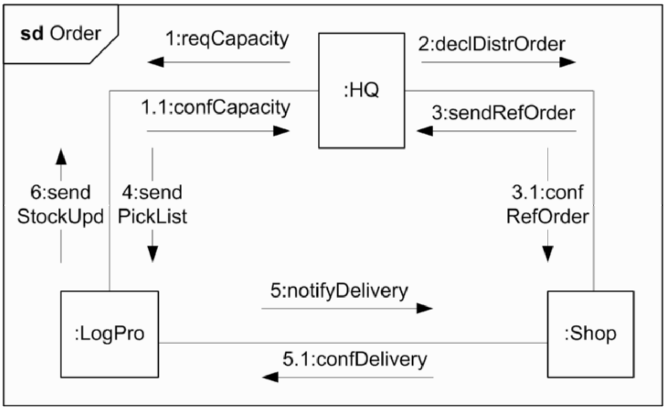


Fig. 5. Communication diagram

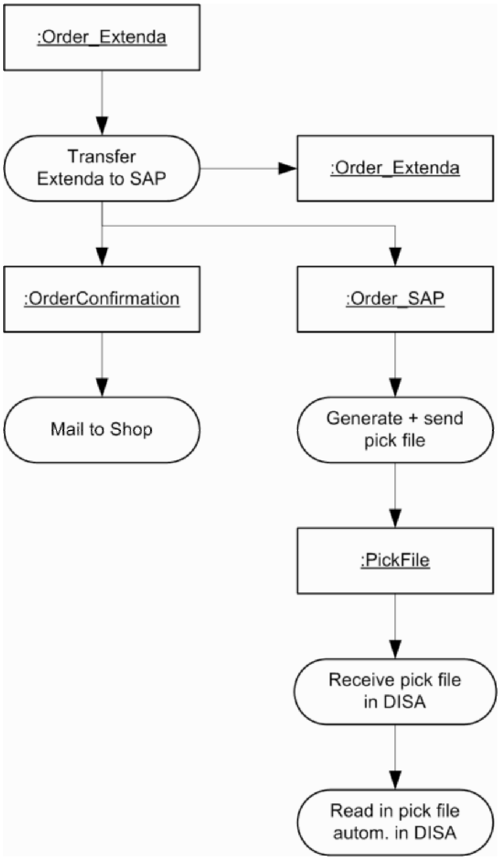


Fig. 6. Activity diagram

All in all the design involved one class diagram, six activity diagrams, one communication diagram and eight state machine diagrams. They were developed by one modeler in 68 hours. In a previous project of similar size the same modeler needed 92 hours without the help of the framework which was not yet developed at that time. These figures seem to indicate that the use of the framework has reduced the effort in that case by 35 %. But this has to be taken with a grain of salt as we cannot be sure that all other parameters were the same in both cases. Further empirical research in that area is therefore necessary.

5 Conclusion

The principal result of our investigation is that the ontological foundations of two particular languages for enterprise modeling and enterprise computing, respectively, exhibit a significant intersection. This common ground allows us to support the translation of enterprise models into computing models in a constructive way. This is done by mapping concepts of the “source language” (here: SIMM) to concepts of the “target language” (here: UML) and by specifying an integration framework that relates the graphs/diagrams of both languages to each other. The latter supports the translation of complete models, diagram by diagram.

We have applied this framework in a project that involved the development of both enterprise models and computing models. We found out that the time for developing the latter decreased by 35 % in relation to time time spent on a similar project without the support of the framework.

References

Ackoff RL (1981) Creating the corporate future. Wiley, New York.

Axelsson K, Goldkuhl G, Melin U (2000) Using Business Action Theory for dyadic analysis. Paper presented at the 10th Nordic Workshop on Interorganisational Research.

Axelsson K, Segerkvist P-A (2001) Interaction between actors and information systems in web-based imaginary organisations – Experiences from two case studies. Paper presented at the 1st Nordic Workshop on Electronic Commerce.

Bunge M (1977) Ontology I: The Furniture of the World. Reidel, Dordrecht.

Bunge M (1979) Ontology II: A World of Systems. Reidel, Dordrecht.

Checkland PB (1988) Soft systems methodology: An overview. Journal of Applied Systems Analysis 15:27-30.

Churchman CW (1968) The Systems Approach. Dell Publishing, New York.

Evermann J, Wand Y (2001) Towards Ontologically Based Semantics for UML Constructs. In: Kunii HS, Jajodia S, Sølvberg A (eds) ER 2001, 20th International Conference on Conceptual Modelling, Yokohama, Japan, November 27-30, 2001. Springer, Berlin, pp 354-367.

Forsgren O (2005) Churchmanian Co-design – Basic Ideas and Application Examples. Paper presented at the ISD 2005 - Information Systems Development, Karlstad, August 14-17, 2005.
Goldkuhl G (1996) Generic business frameworks and action modelling. In: Dignum F, Dietz J, Verharen E, Weigand H (eds) Communication Modeling - The Language/Action Perspective, Proceedings of the First International Workshop on Communication Modeling. Springer, Berlin.
Goldkuhl G (1998) The six phases of business processes - business communication and the exchange of value. Paper presented at the 12th biennial ITS conference "Beyond convergence" (ITS´98), Stockholm.
Goldkuhl G (2002) Anchoring scientific abstractions – ontological and linguistic determination following socio-instrumental pragmatism. Paper presented at the European Conference on Research Methods in Business and Management (ECRM 2002), April 29-30, 2002, Reading.
Goldkuhl G (2005) Socio-Instrumental Pragmatism: A Theoretical Synthesis for Pragmatic Conceptualisation in Information Systems. Paper presented at the 3rd International Conference on Action in Language, Organisations and Information Systems (ALOIS), University of Limerick.
Goldkuhl G, Lind M (2004) Developing e-interactions – A framework for business capabilities and exchanges. Paper presented at the 12th European Conference on Information Systems, June 14-16, 2004, Turku, Finland.
Goldkuhl G, Melin U (2001) Relationship Management vs Business Transactions: Business Interaction as Design of Business Interaction. Paper presented at the 10th International Annual IPSERA Conference, Jönköping International Business School.
Haraldson S, Lind M (2005) Broken patterns. Paper presented at the In Proceedings of the 10th International Conference on the Language Action Perspective, Kiruna, Sweden.
Johansson B-M, Axelsson K (2004) Communication media in distance selling – Business Interactions in a B2C Setting. Paper presented at the 12th European Conference in Information Systems (ECIS), Turku, Finland.
Johansson B-M, Axelsson K (2005) Analysing Communication Media and Actions – Extending and Evaluating the Business Action Matrix. Paper presented at the In Proceedings of the 13th European Conference on Information Systems, Regensburg, Germany.
Lind M, Goldkuhl G (1997) Reconstruction of different business processes - a theory and method driven analysis. Paper presented at the In Proccedings of the 2nd International Workshop on Language/Action Perspective (LAP97), Eindhoven University of Technology, The Netherlands.
Lind M, Hjalmarsson A, Olausson J (2003) Modelling interaction and coordination as business communication in a mail order setting. Paper presented at the 8th International Working Conference on the Language Action Perspective (LAP2003), Tilburg, The Netherlands.
Melin U, Axelsson K (2004) Emphasising Symmetry Issues in Business Interaction Analysis and IOS. Paper presented at the Sixth International Conference

on Electronic Commerce, ICEC'04, Delft University of Technology, The Netherlands.

Melin U, Goldkuhl G (1999) Information Systems and Process Orientation - evaluation and change using Business Action Theory. In: Wojtkowski W (ed), Systems Development Methods for Databases, Enterprise Modeling, and Workflow Management. Kluwer Academic/Plenum, New York.

Mitroff II, Mason RO (1981) Creating a dialectical social science. Reidel, Dordrecht.

OMG (2004) UML 2.0 Superstructure Specification. Retrieved December 20, 2005, from http://www.uml.org/

OMG (2006) Unified Modeling Language: Infrastructure. Retrieved April 18, 2006, from http://www.uml.org/

Opdahl AL, Henderson-Sellers B (2002) Ontological Evaluation of the UML Using the Bunge-Wand-Weber Model. Software and Systems Modelling 1(1):43-67.

Röstlinger A, Goldkuhl G (2005) Grafnotation för SIMM metodkomponenter. VITS/IDA, Linköpings universitet, Linköping.

Wand Y, Weber R (1989) An Ontological Evaluation of Systems Analysis and Design Methods. In: Falkenberg ED, Lindgreen P (eds) Information Systems Concepts: An In-Depth Analysis. North-Holland, Amsterdam, pp 79-107.

Wand Y, Weber R (1995) On the deep structure of information systems. Information Systems Journal 5:203-223.

Weber R (1997) Ontological Foundations of Information Systems. Coopers & Lybrand and the Accounting Association of Australia and New Zealand, Melbourne.

The Integration of Functional Decomposition with UML Notation in Business Process Modelling

Adam Przybyłek

Gdańsk University, Department of Information Systems, 81-824 Sopot, ul. Armii Krajowej 119/121, Poland
adam@univ.gda.pl

Abstract: Over the past decade business and software modelling have been carried out using different notations designed to fit the special needs of the respective tasks. This approach has introduced a gap between the software and business models, which has resulted in inconsistency. The objective of this paper is to propose a new approach to analysis of the business process using UML notation. UML is the most commonly used language in object-oriented software development. Moreover, it can be extended to the modelling of business processes. With this approach it is possible, in the same notation, to visualise a company's business processes, its information system requirements and the architecture of the relevant information system.

However, object-oriented decomposition is not well suited to business process modelling. In an object-oriented methodology the decomposition is achieved by objects on which operations are performed. The aspects of the process are hidden within the objects. In this case functional decomposition from structured methodology is a much better choice. Functional decomposition provides a high degree of abstraction, which is sufficient for business experts. It simplifies the analysis of business by breaking down (decomposing) the problem into the functional steps of which it is made up. The primary aim of this paper has been to integrate functional decomposition with UML notation in order to model business processes. A case study is presented as an illustration.

Keywords: business process modelling, process diagram, Eriksson-Penker Business Extensions, functional decomposition, UML

1 Introduction

In order to survive in today's global economy more and more enterprises are having to redesign their business processes. The competitive market creates the demand for high quality services at lower costs and with shorter cycle times. In such an environment business processes must be identified, described, understood and analysed to find inefficiencies which cause financial losses. One way to achieve this is by modelling. Business modelling is the first step towards defining a software system. It enables the company to look afresh at how to improve organisation and to discover the processes that can be solved automatically by software that will support the business. The outcome of business process modelling is to provide information system developers with the information system requirements. For information system development to be simplified the business model should be drawn up in an understandable notation by software engineers. UML is such a notation. In this paper I show that UML is suitable not only for IT professionals but also for the business community. As a result, UML enables all "stakeholders" to speak the same language. The next crucial issue in business modelling is dealing with the complexity of the business processes. The object-oriented methodology commonly used in software development deals with complexity in concepts of abstraction, encapsulation, generalisation and aggregation. However, object decomposition may not provide the best solution for business process analysis. Object decomposition perceives a system as a collection of collaborating objects, the object being the basic unit by which we organise knowledge. Thinking about a system in terms of messages passing (service requests) between objects is too low a level of abstraction. The natural approach for business people is to break down business processes into smaller sub-processes, as long as an elementary level is achieved that is simple enough to operate. This is called functional decomposition. My approach takes advantage of both UML notation and functional decomposition.

The remainder of this paper first provides a background for business process modelling and functional decomposition followed by a brief introduction to UML and the UML extension mechanism. The proposal, made by Eriksson-Penker Business Extensions and the author, to integrate it with functional decomposition is then examined, after which a case study is presented. The case study shows the application of the proposed approach in a business process model of a Polish institution of higher education in the private sector. The last section of the paper sets out the conclusions to be drawn.

2 Background

The aims of this section are to explain the concepts involved in the business process and to provide an overview of modelling from a business perspective. Then a well-known technique in software engineering, that of functional decomposition is introduced. This section also explains the need for a common language among software developers and business people. Subsequently the characteristics are described that make UML an adequate business modelling language.

2.1 The business process

Numerous definitions of the business process have been proposed in a variety of publications [19, 24, 4, 13, 11, 14]. The business process is a structured collection of activities that work together to produce an outcome in accordance with the business goals. The business process is typically cross-functional and cuts across departmental boundaries. Whether manual or automated, the process takes inputs, transforms them, adds significant value to them and provides outputs for its customers. The customer may be either internal or external to the business.

A resource is an object which can play a role in the realisation of a process. Resources can be classified as follows: input object, output object, supplying object and, controlling object [8]. The input objects are the resources needed to perform the process. These are usually transformed or consumed as part of the process. The input objects can also be refined by the process, in which case the process adds value to them. The output objects represent the accomplishment of the goals and are the result of the process. The output object may be a completely new object created during the processes or it may be a transformed input object. The output of one business process is often the input to another process. The process may also have connections with the supplying and controlling objects. The supplying objects carry information required by the process. Resources, such as people, that are responsible for executing the activities in the process are known as control objects.

Each business process has a goal that reflects business expectation of its outcome. All the processes collaborate to achieve the strategic goals of the business. The process may be defined at any level of viewing from enterprise-wide processes to a single task performed by a single person. The process can consist of lower level sub-processes. The lowest level of the process is referred to as the elementary business process (EBP), a process performed in one location at one time which leaves the business in a consistent state [11].

The main types of business process are core (primary) processes, support processes and management processes [8, 16]. The core processes are those that define the purpose of the existence of the organisation and contribute directly to the production of goods and services for the organisation's external customers. The support processes help to fulfil the services of the main processes. Their outcome is not visible to the external customer but is essential to the effective functioning of the business. These processes make the delivery of the core processes of an enterprise possible. The last category of process, the management process, describes the work of managers in support of the business processes. These processes affect the way the other processes are managed and the relationship of the business to its owners.

2.2 Business process modelling

Business processes are complex, dynamic and interconnected throughout an organisation, sometimes extending to its customers and deliverers. Modelling techniques have been created to deal with this complexity. The model demonstrates the relationships between processes and provides an understanding of how the business operates. Models allow irrelevant details to be eliminated so that the focus can be on important aspects of the business. The business model is a mechanism for capturing fundamental enterprise business knowledge and represents the essence of the business organisation. It describes at a high level of abstraction how the business is working today and how this might be improved upon tomorrow. The business model generates an overall picture of the business domain and enables the effects of changes to be identified before the changes have been implemented.

B. Baker writes [1] that there are three basic reasons for modelling a business: to re-engineer the business, to improve a business process or to automate a business process. The goal of business process re-engineering is radical re-organisation of an enterprise along the flow of work that generates the value sought by the customer [11, 14]. To achieve this goal the business processes must be revised, investigated and documented. This will enable the owners of the business processes to gain a better understanding of enterprise-wide operations. The next step is to find bottle-necks in current processes and to determine how to make necessary improvements so that the processes are more productive and cost-effective. The last reason for business modelling is to take decisions about which of the manual processes might be automated and conducted better by information systems. Automation of business processes is the primary use of information technology in organisations. It reduces human intervention and assembles software services to support the business. A clear map of the business processes at the beginning of a project helps to specify requirements for information systems and ensures that these systems meet the business needs when delivered.

2.3 Functional decomposition

The term “functional decomposition” was introduced with structured methodology and was the key concept in software engineering prior to object-oriented domination. The first stage in most of the system’s life-cycles is business planning. Before analysts begin developing information systems they must understand the major business functions that the organisation needs to perform. It is then much easier to identify processes that occur within the business functions and the system that will ultimately support those processes [12].

A commonly used technique in business analysis is functional decomposition. This is a top-down analysis that views a business as a collection of functions. Each function can be decomposed into business processes, each of which may be progressively broken down into smaller sub-processes until the business has been specified in terms of EBPs. These can once again be broken down into actions.

The order in which these actions are performed and the flow control can be presented at an activity diagram, as explained in a later section.

2.4 Unified Modelling Language (UML)

Software development is known to be a complex task requiring the co-ordinated efforts of domain experts, business analysts, designers, programmers and testers. In the IT community UML notation is considered to be the industry's standard for object-oriented system development and is widely used all over the world. Different phases of the software development process require different numbers of diagrams, where each diagram emphasises a particular aspect of the system. UML 2.0 defines 13 types of diagrams which are divided into two categories: structure diagrams, and behaviour diagrams [26, 29]. All the diagrams put together create a complete picture of the system. For more detailed information about UML diagrams please refer to the information sources mentioned in the reference sections [26, 9, 21, 10, 29, 27, 28].

Nevertheless, there is no one standard in business modelling and various notations exist [13, 20, 18, 7, 6]. As emphasised in [1], however, there is a need to use the same notation throughout the entire life-cycle of a project. Such an approach provides a great advantage for all stakeholders, making the development process more efficient. By having the business experts and system developers using the same modelling concepts, the risk of costly errors related to misunderstanding of the models is significantly decreased. When the issues mentioned above are taken into account, UML is a good candidate as a business modelling tool [4, 20, 23, 17, 22, 25]. It provides a common notation that allows business people to express the design of their business processes in a way meaningful to them, while providing clear direction for information technology support.

UML was initially designed as a general-purpose visual modelling language used to specify, visualise, construct and document the artefacts of a software system [26]. However, its clear and easily comprehended notation makes UML understandable to non-technical stakeholders. A flexibility and extensibility meta-model enables UML to be extended into different areas of modelling, including business process modelling.

2.5 The UML extensibility mechanisms

No language could ever be adequate to express a specific subject across all domains. UML is designed to be open-ended, allowing users to customise and extend it to their own particular needs [2]. The UML extensibility mechanisms refer to stereotypes, constraints and tagged values [26, 9, 21]. A stereotype defines a new type of modelling element suitable for a specific problem. Each stereotype must be based on a certain existing type in the meta-model. A stereotype may be expressed as a label enclosed in guillemots «» or as an icon. Constraints represent restrictions that are applied to UML elements. A constraint is a text enclosed in curly brackets {}. Constraints may be expressed more formally using the Object Constraint Language (OCL). Tagged values are properties attached to UML elements as a name-value pair. Both the tag and the value are enclosed in curly brackets {tag = value}. The UML extension mechanism should be used in those cases in which the basic semantics of the language is insufficient.

3 Applying UML in business process modelling

This section presents two of the most useful diagrams in business process modelling: the activity diagram and the process diagram. The activity diagram is a useful tool for showing flow control in the outworking of a business process. The process diagram is part of the Eriksson-Penker Business Extensions and is suitable for gaining an overview of all the processes.

3.1 The activity diagram

The most important UML diagram from a business point of view is the activity diagram. This combines ideas from the event diagrams of Jim Odell, data flow diagrams and Petri nets and is the latest form of the flow chart. The flow chart, which treats a process as a collection of actions, is one of the most natural visualisations of a business process and makes the activity diagram suitable for the specification of business processes [20, 18, 6, 25, 3, 5, 15].

The activity diagram can describe complex business processes as well as elementary business processes. A functional decomposition approach organises activity diagrams in a hierarchical tree, depending on the complexity of the business under study. Each complex process is modelled as a sequence of activities that represents subprocesses. The elementary business process is modelled as a sequence of atomic actions. Too many activities on the same diagram will cause the diagram to become confusing. Functional decomposition provides a way of preventing confusion by creating diagrams at different levels of abstraction and organising them hierarchically. Activity from one diagram can be divided into actions in a lower level diagram.

Activities are presented as ovals and are governed by clearly defined control nodes such as decisions and merges (diamonds icon), forks and joins (solid horizontal line). When an activity is completed the control flow is passed to the next activity. A transition is modelled as a directed arrow from the source activity to the target activity. Objects may participate in the activity. They are either an input to or an output from the actions, or they can simply show that an object is affected by a specific activity [9]. Object flows are shown with a dashed line ending with an arrow. Partitions can be used to show in which part of an organisation the activity is performed. Moreover, the activity diagram can present more than one business process. However, if there is a need to give a broad picture of all the processes, the process diagram is a better solution. I also suggest using the process diagram instead of the activity diagram when objects involved in activity are crucial and there is a need to emphasise them.

3.2 The Eriksson-Penker Business Extensions

A number of techniques have emerged over the years to support business modelling, such as IDEF0, IDEF3, ARIS-EPC, BPMN, and Petri nets. These provide ways of expressing business processes in terms of activities and of arranging resources. None of them, however, is equipped to support the design of software. This creates a gap between business and software models and is a source of mistakes.

H.E. Eriksson and M. Penker have proposed extensions to UML that are intended as a basic framework for business modelling [24, 20]. These extensions can express a business process while at the same time providing consistency with a software model. Eriksson-Penker Business Extensions use four different views of a business: business vision, business process, business structure, and business behaviour. Each view focuses on a particular aspect of the business and is described through a number of diagrams and/or a textual document [8]. For the purpose of this paper only the process diagram, which is an element of the business process view, will be presented.

3.3 The process diagram

The process diagram is a specialisation of the UML activity diagram with a set of stereotypes that illustrates the interaction between the processes, the resources that participate in the processes and the goals of the processes. The process diagram describes how work is done within the business environment. The following stereotypes are used to define the process diagram: business process, resources, goals and rules.

The business process stereotype

The core business modelling element is represented by stereotyping an activity to a «process» or by a special icon. High-level processes may be divided into subprocesses placed inside the process element or on a separate lower level diagram. I suggest breaking down business processes until EBPs are achieved. Each EBP can then be specified on an activity diagram by internal actions. In this manner we obtain functional decomposition in UML notation.

Resource stereotype

The resource types are represented as classes. Resource instances are represented as objects. Input objects are usually placed on the left of the process and output objects are placed on the right. A result produced by one process can constitute input to another process. The supplying and controlling objects are placed below or above the process and can be linked to it by a dependency stereotyped adequately by «supply» and «control». Eriksson-Penker Business Extensions [8, 9] define the hierarchy of resource types (Figure 1). In the process diagram the resource type is indicated by the stereotype. The stereotype provides additional information about the resource. Abstract objects refer to intangible items. The "people resource" denotes human beings acting in the process.

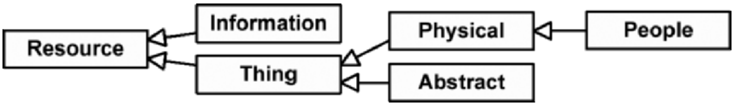


Fig. 1. Resource types

4 A case study

In this section, the demonstration that the proposed approach is capable of modelling business processes, is presented. For the purpose of the illustration, a business process model of an institution of higher education was developed. The research method applied to this organisation was open-ended interviewing. The persons who are most knowledgeable about the issue of interest were interviewed. Relying on the information collected from the interviews, a textual description to illustrate how the business is run was created. The first part of this section gives an overview of the main business processes, although the details are beyond the scope of this paper. The second part presents the relevant process and activity diagrams. Functional decomposition was applied to break the business processes into atomic actions.

4.1 Domain specification

The business domain modelled is a Polish institution of higher education in the private sector. The academic institution can be understood as a service provider, its main business function being the provision of higher education. This service is immaterial and students participate in it directly. The main business function is achieved by nine high level processes. The overview of each process is presented below as a textual description.

Analysis of business environment

Heads of departments are continuously tracking changes in the job market, to adjust what is offered to the needs of the educational market. IT magazines and internet portals are an essential source of this information. These members of staff are also watching the micro-environment, other colleges with which they are competing on the student market. The data for their analysis is gained from the internet web sites of other institutions of higher education. One of the main areas of interest is therefore what other educational institutions have on offer. The college's educational programme is drawn up on the basis of such data. The whole process is supervised by the Chancellor and verified by the school's position in the student market.

Marketing

The educational programme thus established has to reach a wide range of customers. In order to achieve this administrative staff create marketing materials tailored to the educational programme. These materials are disseminated through the press, radio and TV. Administrative workers additionally compile the contents of the website. Website creation is handled by an external company.

Human Resources Management

Heads of department have to select academic staff in the light of the educational programme offered. First the teaching of the academic courses is opened to those lecturers who were employed in the previous academic year and who were highly rated by students. Where new lecturers have to be employed, the background of candidates' research work is taken into consideration. Next payment is negotiated individually with each lecturer and the interest rate and staff profiles are accepted by the Chancellor. The assignment of lecturers to subjects is set out in a document known as the workload card, which must be signed by the Chancellor.

Teaching arrangements

Once classes have been assigned to the lecturers, the next step is the preparation of syllabuses. The lecturers can follow syllabuses that were used during the previous year and custom them to current scientific developments. The syllabuses must be approved by the Dean.

Ordering books

On the basis of the syllabuses, lecturers' advice and students' requests administrative staff draw up order forms for the books for the school library and monthly orders are sent to the publishers.

Timetabling

On the basis of the workload cards and classroom availability the timetable is drawn up by administrative staff, who contact the lecturers to find out when they are free to teach.

Conducting the classes

During the academic year lecturers conduct their classes according to the timetable. Consequently the students make educational progress.

Assessment of lecturers

After each semester the Assistant to the Dean ranks the lecturers on the basis of surveys, inspections and the attendance register. The surveys are conducted among the students by the Assistant to the Dean, the inspection is carried out by the Head of Department and the attendance register is kept by the porter.

Admission procedures

All candidates must submit to the Dean's Office an application form and a copy of their final school examination certificate. If the number of applicants exceeds the number of places available, the applicants undergo a process of evaluation based on their scores in their final school examination and an entrance examination. Applicants are notified of the exact time of their entrance examination and their qualifications are judged by the admission staff. If the number of applicants is lower than the number of places available, all applicants are accepted. Accepted applicants receive a letter asking them to pay their tuition fee, and the list of students admitted is drawn up according to the payment receipts.

4.2 Business process model

In this part the use of the activity and process diagrams to achieve functional decomposition was demonstrated. Figure 2 shows the business process in the organisation analysed in its entirety.

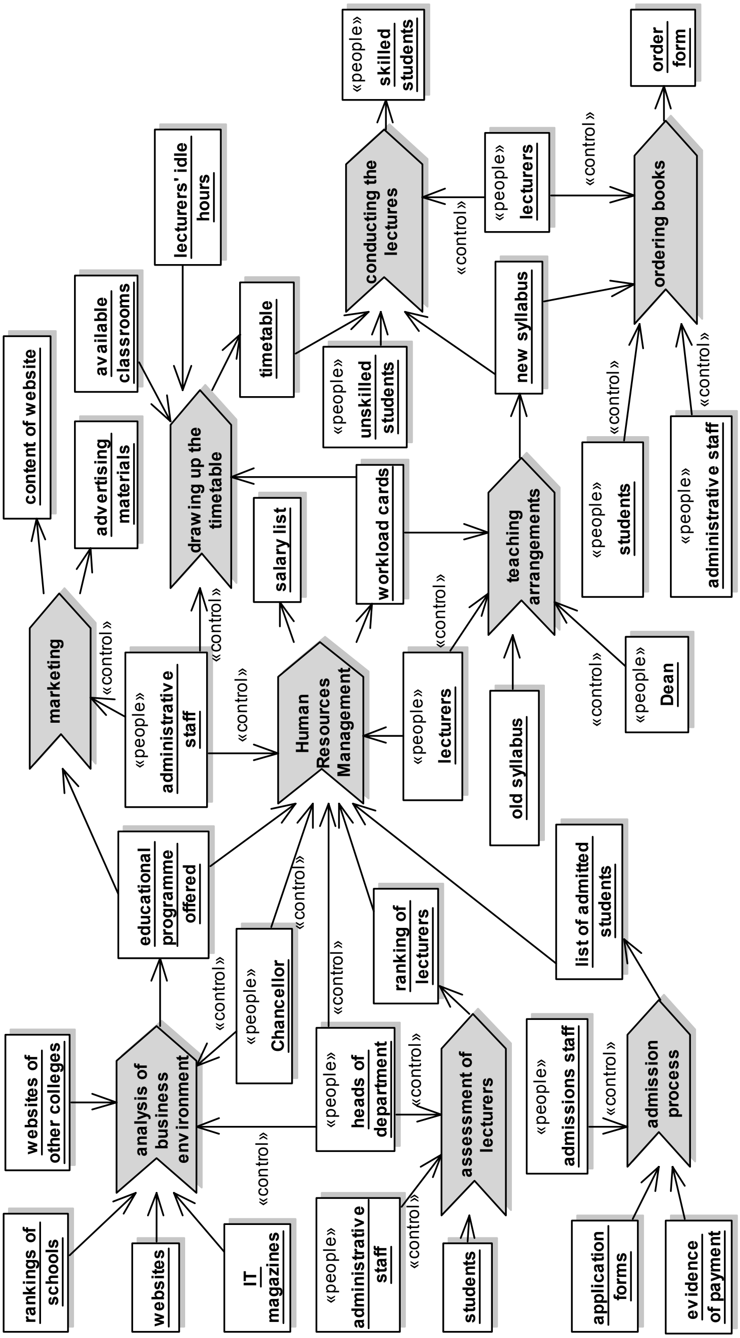


Fig. 2. Providing higher education

The goal of providing higher education is to prepare students to apply scientific knowledge and methods in their profession. The figure 2 conveys general information. The assessment of lecturers and admission procedures are given further attention. The admission of students to courses is an example of an EBP and so I have used an activity diagram to present it (Figure 3).

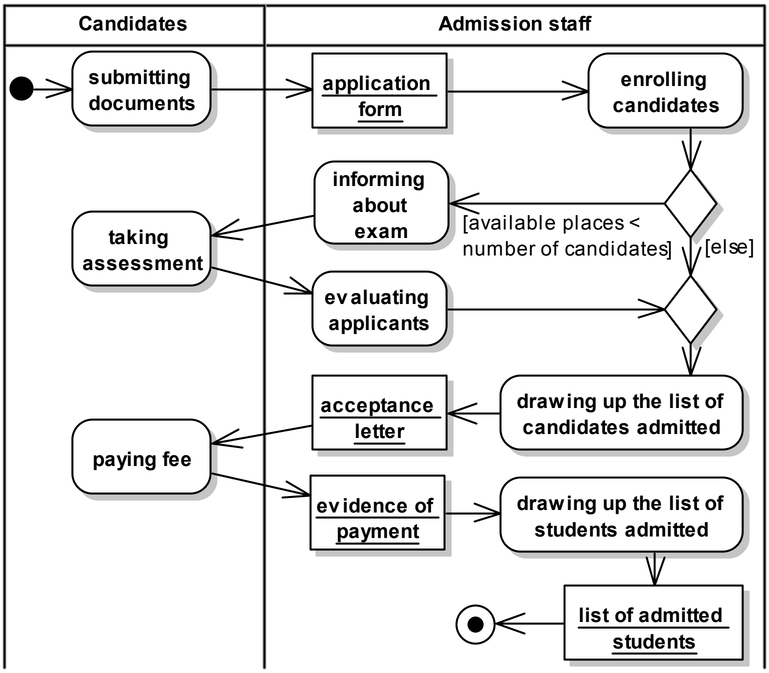


Fig. 3. Admission process

Figure 4 illustrates the four EBPs that make up the assessment of lecturers.

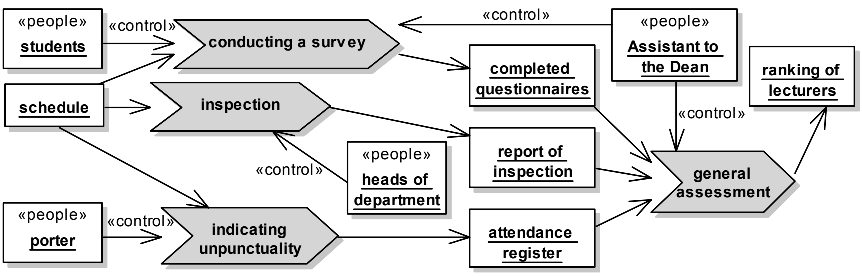


Fig. 4. Assessment of lecturers

In order to analyse the process of conducting the survey functional decomposition has been applied once again. The result is shown in Figure 5.

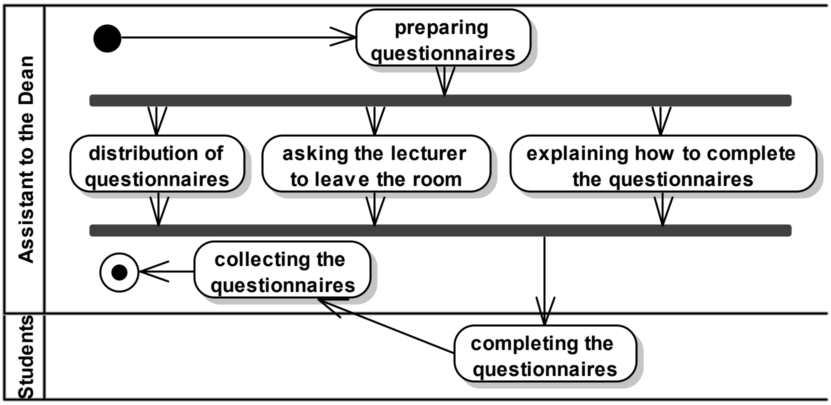


Fig. 5. The process of conducting the survey

In order to gain deeper understanding all business processes should be decomposed and shown in detail in the activity diagram.

5 Conclusion

Business modelling has appeared as a response to the need for describing a business in a formal way. The purpose of creating a formal model is to better understand, analyse, improve or replace a business process. In the last decade business practitioners have been overwhelmed with notations in diagramming business processes. These notations have been understandable for business people but have not been compatible with the notations used in software engineering.

The research contributes to business modelling in several ways. First of all, the proposed study introduces a new approach to business process modelling. It integrates a well-known strategy for dealing with complexity, that of functional decomposition, with a notation commonly used in software engineering, namely UML. The case study suggests that such integration is possible, but this should be confirmed empirically. Functional decomposition offers the analyst a mechanism to divide and conquer the complexity of business processes. The UML extension mechanisms allow UML to be fitted for specific needs. It is notation that can be used for business modelling just as it has come to be used for software modelling.

Secondly, the presented approach guarantees a smooth transition from business process analysis to system requirements, because the same notation is used for all models. This significantly decreases the time needed to transform the model from a business to a software model. Moreover, the availability of a common notation, aligning IT with the business community, improves communication among a project team. Good communication is a critical success factor in the development of information systems.

References

Baker B (2001) Business Modeling with UML: The Light at the End of the Tunnel. Rational Software

Booch G, Rumbaugh J, Jacobson I (1998) The Unified Modeling Language User Guide, Addison Wesley

Bruno G, Torchiano M, Agarwal R (2002) UML Enterprise Instance Models. CIT 2002

Castela N, Tribolet J, Silva A, Guerra A (2001) Business process modeling with UML. ICEIS 2001, pp 679-685

Desfray P (2004) Making a success of preliminary analysis using UML, Softeam

Dewalt C (1999) Business process modeling with UML. Johns Hopkins University Press

Eertink H, Janssen WPM, Oude Luttighuis PHWM, Teeuw WB, Vissers CA (1999) A Business Process Design Language. Lecture Notes In Computer Science, vol 1708, Springer

Eriksson HE, Penker M (2000) Business Modeling with UML: Business Patterns at Work. John Wiley & Sons, USA

Eriksson HE, Penker M, Lyons B, Fado D (2004) UML 2 Toolkit. Wiley Publishing, Inc., USA

Fowler M, Scott K (1999) UML Distilled – A Brief Guide to the Standard Object Modeling Language. 2nd ed., Reading, MA

Frost S, Allen P (1996) Business Process Modeling, SELECT Software Tools plc, Cheltenham

Harel D, Politi M (1998) Modeling Reactive Systems with Statecharts. McGraw-Hill, New York, USA

Havey M (2005) Essential Business Process Modeling. O'Reilly, USA

IBM Corporation (1996) Business Process Reengineering and Beyond, California

Loos P, Fettke P (2001) Towards an Integration of Business Process Modeling and Object-Oriented Software Development. Chemnitz Univeristy of Technology, Germany

May M (2003) Business Process Management. Pearson Education Limited, Great Britain

McLeod G (1998) Extending UML for Enterprise and Business Process Modeling. UML'98 - Beyond the Notation, First International Workshop, pp 195-204

Odeh M, Beeson I, Green S, Sa J (2002) Modelling Processes Using RAD and UML Activity Diagrams: an Exploratory Study. ACIT 2002

Omar A, Sawy EL (2001) Redesigning Enterprise for e-Business, McGraw-Hill

Ovidiu SN (2000) Business Modeling: UML vs. IDEF. Griffith University

Pender T (2003) UML Bible. John Wiley & Sons, USA

Podeswa H (2005) UML for the IT Business Analyst: A Practical Guide to Object-Oriented Requirements Gathering. Thomson Course Technology PTR, Canada

Sinogas P, Vasconcelos A, Caetano A, Neves J, Mendes R, Tribolet J (2001) Business processes extensions to UML profile for business modeling, Centro de Engenharia Organizacional, Lisbona. ICEIS 2001, pp 673-678

Sparks G (2000) An Introduction to UML: The Business Process Model. Sparx

Ståhl H, Jankko T, Oittinen P (2002) Modelling of publishing processing with UML. Helsinki University of Technology, Department of Automation and System Technology, Laboratory of Media Technology, Finland

UML 2.0 Superstructure Specification (2003) Object Management Group, Inc.

Wrycza S, Marcinkowski B (2005) UML 2 Teaching at Postgraduate Studies - Prerequisites and Practice, The Proceedings of ISECON 2005, vol 22, New Orlean

Wrycza S, Marcinkowski B (2005) Interaction Occurrences and Combined Fragments in System Dynamics Modelling with UML 2 Sequence Diagram art. (w:) (ed.) Nilsson G., Gustas R., Wojtkowski W., Wojtkowski G., Wrycza S., Zupancic J., ISD 2005 Proceedings of the Fourteenth International Conference on ISD, Karlstad University Studies, Karlstad, pp 59-68

Wrycza S, Marcinkowski B, Wyrzykowski K (2005) UML 2.0 in information systems modelling. Helion, Warsaw

The Framework for Adaptable Data Analysis System Design

Olegas Vasilecas, Aidas Smaizys

Department of Information Systems, Klaipeda University, Lithuania.
olegas@isl.vtu.lt, aidas@oil.lt.

1 Introduction

Enterprises today own large databases (DB) and widely use information systems (IS). IS used in business are mostly dedicated to various data input and analysis. The result of data analysis is information contained in specially designed reports, OLAP cubes, etc. Such information is used to track and evaluate the business situation according to the business strategy and tactics and determine conformance to the business policy and rules [10].

The decision making action is one of business system processes which follows rules too. Usually the operator of the information system is responsible for decision making and the part of business system is dedicated for data analysis processes to support such decisions [12]. The sources of data analysis rules are all the documents, legal requirements and laws [1], mutual agreements, business culture, limitations in various resources (people, software systems, hardware, etc.) and architecture, but most valuable of them are – experience and knowledge of the employees. Sometimes such knowledge or part of it is embedded into reports or stored in various knowledge bases in enterprises, but mostly it is lost after change of the staff.

According to the traditional system engineering the automation of business processes is done by transforming business rules (BR) into the functional requirements of platform independent information system and translating it into the specification of platform dependent software system (SS) [13]. The specifications are used later for coding of the final applications. Following such process the business rules are embedded into code and lost forever [1]. The same is with the analysis rules which are embedded in the traditional reporting and analysis SS. Part of the captured BR in functional requirements are data analysis rules already or are transformed into data analysis rules later. Although BR stated as requirements are stored and transformed statically and can not be changed later without all redesign cycle.

Another big flaw is pre-programmed in traditional design processes from the very beginning – it is different and mostly false interpretation of business rules stated as requirements or transformed into specifications by analysts or designers and into final software code by programmers. This entire staff directly involved in software application design and development is not involved in business directly. Such misinterpretations are replicated through all the design cycle and sometimes are not discovered even in testing phase, because of testing scenarios based on the same requirements. It can not be discovered even in already deployed software until some specific interference of various factors arises. It is very dangerous in large critical systems, because the corrections can not be done quickly and the whole redesign process is needed.

In this paper we discuss the possibility of use of the BR represented in XML for intelligent adaptive dynamic data analysis and representation of the information by automation of recurring decision making processes and describe the framework for such software system design. The paper states the goals of data analysis using BR approach and possible ways for solutions using XML transformations from business rules into dynamic executable MDX instructions used for data analysis and representation of the information.

The rest of the paper is composed as follows: Sect. 2 analyzes related work on system engineering methods used for data analysis, information processing and decision support, Sect. 3 describes enterprise material and information flow model, Sect. 4 proposes business rule based framework for adaptive data analysis system design, Sect. 5 presents experimental evaluation of the framework and the final section summarizes the proposed method.

2 Related works

The main difference between traditional data analysis methods and business rule driven is that the rule based data analysis involves business logics into data analysis processes using transformations of business rules into different artefacts in IS and SS instead of implementing them directly in the requirements or design specifications and SS code.

From the information system perspective, "...a business rule is a statement that defines or constrains some aspect of the business. It is intended to assert business structure, or to control or influence the behaviour of the business." [2] There are other business rule (BR) definitions as well [3, 5]. Business rules are derived from business policy formulated by business strategy and tactics defined in enterprise mission. Implementation of BR empowers to achieve the objectives [11] and goals setup by executive staff and stated in enterprise vision. Due to dynamics of its nature, business environment is changing frequently according to internal and external influences such as changes in law, new competition etc. Business requires immediate and adequate reaction to changes. Otherwise there is a big risk to fall off competition. This is the main reason not only for immediate analysis of the situation and decisions, but also for the need of continuous changes in business policy and business logics. Such changes challenge related changes in information and software systems.

To allow such functionality in the other papers [14, 15] we have discussed an approach to capture business rules separately and store them in some repository by creating rule model. In this rule model all the business rules, captured in business system, are later transformed into information processing (information analysis) rules in information system. Using such rule model information processing rules are still platform independent and have direct relations to the business rules and business objects and by tracking all the transformations they can be multiple reused in different software systems. Even if not rule based – by transforming such rules into software procedures or other components in particular software system.

Data analysis is one of the main sources of information for prediction of business system changes, evaluation of influence, risk analysis and decisions. Information needed for decision support is derived from captured data using available knowledge. Knowledge can be represented by business rules as well. Data analysis results can be a reason for business changes and influence the change of all the business system or some part of it [11].

Summarising previous results, we can state that two main components for data analysis are needed – data and rules for evaluation and transformation of the data. Such rules operate in business environment as various instructions for data manipulation, business models, policies and laws, business conventions or are derived using knowledge and experience of the operator (human).

In [8, 9] the authors state that by selecting and combining the most suitable and economical web services, business processes can be assigned dynamically by observing the changing business conditions. We think the same way can be used for dynamic data analysis process generation according to the business situation. There are two different ways of data analysis process creation – design of different executable processes in software system, at the same time mapping different processes to different conditions stored in the software system rules and the other one – storing business rules and transformations needed for data analysis process generation and compiling the process, using stored transformations, on the fly according to the current business rule set loaded in the knowledge base of the system.

Experimenting with different methods of business rule representations we have discovered that the business rules represented in XML are easily transformed using specially designed XSLT transformation schema into data analysis SS code, queries and other artefacts according to the properties of rules or rule sets [15]. We have found very suitable use of such transformations for rule based data analysis.

Following proposed method of rule transformation using XSLT schema and according to the principles stated by C. J. Date [4] the rules can be represented in XML and combined into rule sets by rule sources and destinations. All rule sets should be stored in repository. Every new rule should be added to the existing rule set only after checking consistency of the rule set using inference engine first. Complete rule set is passed to the system and transformed into the SS code used for data analysis. On the other hand the changing business situation can create new facts added into the knowledge base of the inference engine making current rule set inconsistent and showing that the revision of the existing rules is needed. Using this approach it is possible to generate smart data analysis reports on-line and react to business system changes. Such reports would use rules for evaluating current business environment and act according to the situation. For example such smart report can display main summary of the business system parameters if there is good business situation (profit greater than 10%) and display detailed information (use slice and/or dice methods according to the OLAP systems, etc.) for such parameters below predefined in the business rules.

3 Enterprise information flow model

Summarising the information analyzed we have created business system material and information flow model (see Fig. 1). The proposed model represents information flow from the data source in material processing layer to the decision making and implementation of decisions. All the data carriers and transformation processes are grouped into four columns by origin according to the method described in [11].

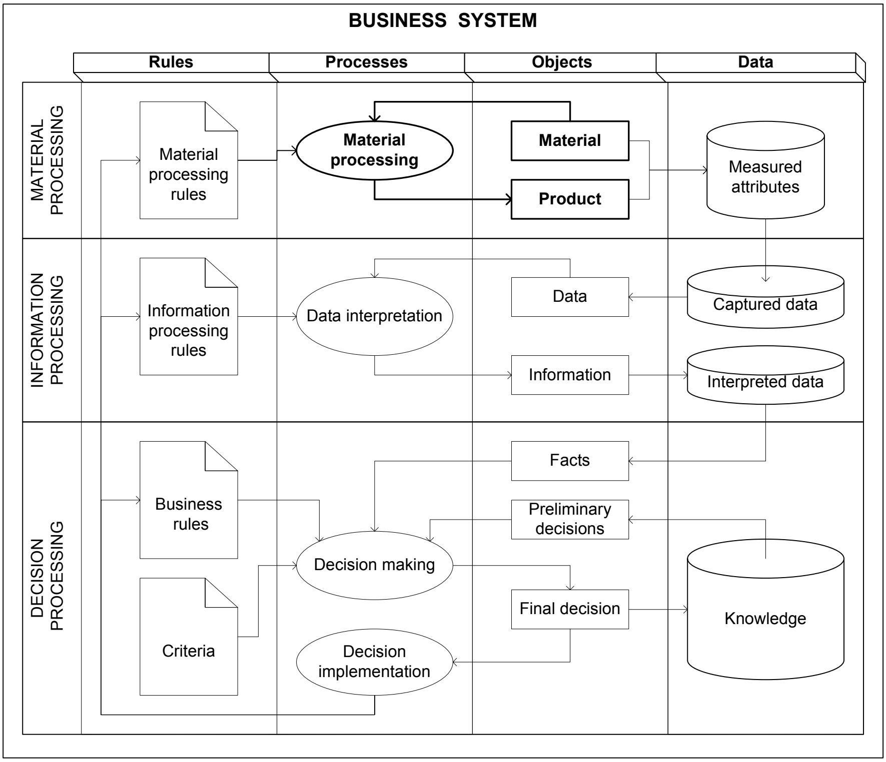


Fig. 1. Business system material and information flow model

In the first material processing layer we have displayed Material processing rules, Material processing process and the processing objects (initial material and final processing product). In business it represents some machinery equipment, for example equipment for production of the windows from the wood. In this case the measured attributes displayed will be the quality of the wood, size, price, etc. In the same way Material processing layer can be treated for services, too. In this case there should be service rules (discount, term) instead of material processing rules and the service process instead of material processing. The input and output objects in this case will be the initial and the final data or state of the maintained objects.

The second layer is dedicated for information processing. It is usually done in several information systems or at least hand written journals. It is represented in the model as captured data. Some captured data are usually used for data interpretation to get summarized information. It is usually done by creating reports (displayed as interpreted data) by predefined information processing rules implemented in the information systems usually using relational databases and SQL queries, but it can be done manually, too.

The third layer is dedicated for decision processing – decision making and implementation of decisions. According to the proposed model decisions are made using facts (information from interpreted data), BR (legal regulations, instructions, business policy, etc.), preliminary decisions based on the knowledge and selected decision criteria. The final decision is the source for new knowledge. The final decision usually is implemented changing material processing or information processing rules. The decision implementation by changing BR is possible, too, but it is out of our investigation scope.

The first two layers are usually easily automated using the present software engineering techniques. The software systems implemented in the business system are dedicated to support decisions of the management staff. However there is a lot of recurring decision tasks which should be automated. Due to the large uncertainty and frequent changes in business system it is still a very problematic and complicated task.

4 The framework

This section deals with automation of decisions according to the proposed enterprise information flow model (see Fig. 1), described in the previous section. For solution of such problem we propose the special software system architecture (see Fig. 2) and realization of the information interpretation and decision processes using special software system components.

First of all the automation of the data interpretation process is needed. Usually it is done by using DBMS built in services, although we have used data analysis services for execution of the MDX instructions dynamically created in the next layer [16].

In the second layer of the software subsystem according to the proposed framework the automation of the decision making and implementation processes is needed.

Decision implementation process is automated using transformations of the proved complete rule set, represented in XML and predefined XSLT transformation schema assigned to the fired preliminary decision, into executable MDX instruction later passed into data analysis service in the upper layer.

The preliminary decisions are validated using logical processing engine implemented in the inference subsystem. Decisions are validated according to the business situation represented by the facts received from interpreted data and predefined business rules. The decision selection process is implemented into the same inference subsystem and the additional criteria are used for selection of the rules which are placed for logical processing or for selection from several valid decisions.

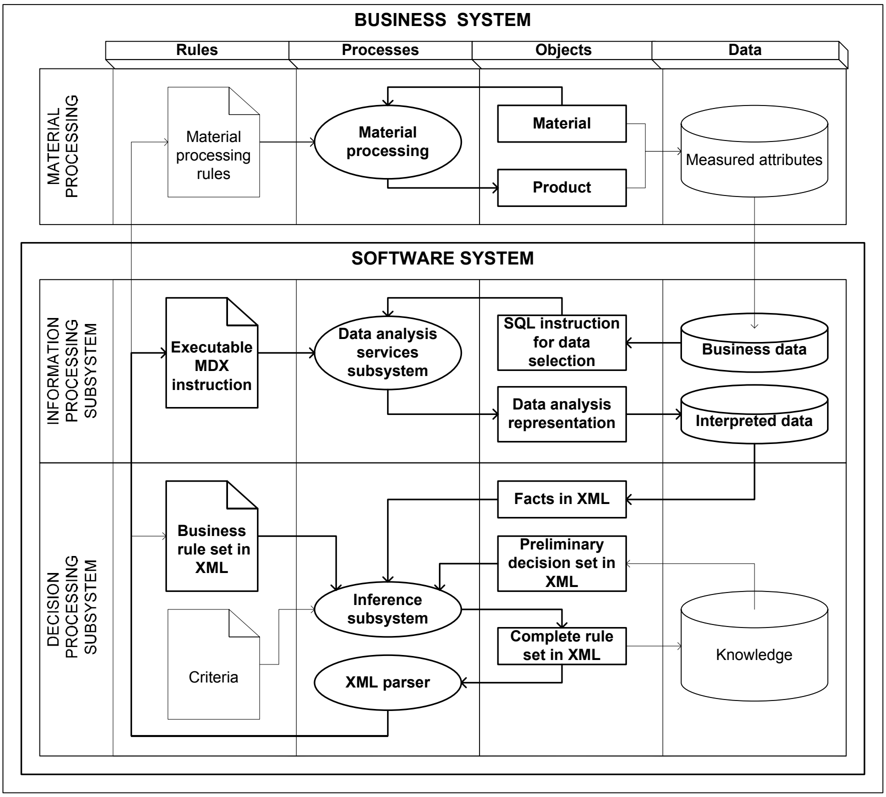


Fig. 2. Software system architecture for adaptable data analysis

According to the proposed framework and software system architecture execution of created MDX instruction the software system will produce the adaptive data analysis according to the current business situation (represented by facts) and business policy (represented as business rules) involved in the inference subsystem.

5 Experimental evaluation of the framework

The framework is still in the experimentation phase. We have made experiments by creating the prototype software system represented in the decision processing subsystem layer of the framework to examine possibility of dynamical creation of the MDX instructions for OLAP based data analysis cubes and pass it into upper information processing layer [16].

We have used the inference engine with backward chaining algorithm for logical processing in inference subsystem. We have selected backward chaining because it is useful in the situation where the quantity of data is potentially very large and we actually need only to prove the validity of the predefined preliminary decisions expressed as hypothetical preliminary decisions in the proposed framework. In case more than one hypothetical preliminary decision is involved, backward chaining attempts to verify each one independently.

We have represented Business rules in XML and transformed them according to the predefined XSLT transformation schema into predicates for use in inference engine and added automatically discovered facts, using predefined SQL queries, to the knowledge base of inference engine. According to the framework such facts represent current business system situation. The final OLAP cube is changed dynamically according to the inference results and is adapted to the current business situation.

During experimentation we have discovered that there are only few possibilities to influence decisions using OLAP cube transformations. Currently we can influence formatting and dimensions of data representation. For example, if the total revenue per moth is less than 10% - detail data are displayed for day-by-day analysis all around the stores; otherwise only summary data is proposed. However the use of inference subsystem and handling of the inference results makes a lot of problems not solved yet.

6 Conclusions and future work

The practical experimentations using proposed framework for adaptable data analysis software system architecture and design have led to the following conclusions:

- Business rules and facts about current business situation represented in XML can be processed in specially designed decision processing subsystem to choose predefined preliminary decision represented in XML.
- If business rules according to the selected preliminary decisions and selected facts make complete set they can be transformed using predefined transformation XSLT schemas to the executable MDX instruction for further use in adaptable data interpretation.
- The information derived from data analysis results can be translated into new facts and entered into the knowledge base of inference engine to evaluate the conformance of the current business system state according to the business policy expressed as business rules in repository and identify the source of inconsistency.
- Rule set and facts stored in the knowledge base of inference engine can be used for what-if analysis by modelling and adding new rules into rule set or adding new facts for examination. By checking consistency the influence of such business system changes can be examined.
- Business rule approach for rule based data analysis allows business rules to be stored separately by creating special SS infrastructure and allowing rapid business rule changes without changing SS code and redesigning all the entire IS. The proposed SS architecture allows reuse of the rule sets and transformations.

The proposed framework and SS architecture can be used for creation of the intelligent adaptable data analysis system. However, performance problems can arise when loading large rule sets into the knowledge base for inference. We expect such problems and limitations to be solved by evolving methods of logical derivations and new hardware.

References

1. Barbara von Hale. Business Rules Applied: Building Better Systems Using the Business Rules Approach. John Wiley & Sons, Inc., New York, 2001.
2. Business Rules Group. Defining Business Rules: What are they really? 3rd ed., 2002, URL: http://www.BusinessRulesGroup.org
3. Business Rules Group. Defining Business Rules: What is a Business Rule?, 2004, URL: http://www.BusinessRulesGroup.org/brgdefn.htm
4. Date C. J. What Not How: The Business Rules Approach to Application Development. Addison-Wesley Longman Inc., 2000.
5. Gladys S.W. Lam. Plainly Speaking: What Is a Rule? Business Rules Journal. May 2003, Vol. 4, No. 5.
   URL: http://www.BRCommunity.com/a2003/b149.html
6. Grosof B., Poon T. SweetDeal: Representing Agent Contracts with Exceptions using Semantic Web Rules, Ontologies, and Process Descriptions.

International Journal of Electronic Commerce, Accepted Journal paper, revised version of Nov. 19, 2003
7. Mahajan S. Building a Data Warehouse Using Oracle OLAP Tools. ACTA Journal, September 1997.
8. Orriens B., Yang J., and Papazoglou M. P. “Model driven service composition”, in: Proceedings of the First International Conference on Service Oriented Computing, Trento, Italy, December 15-18, 2003.
9. Orriens B., Yang J., and Papazoglou M. P. “A Framework for Business Rule Driven Service Composition”, Lecture Notes on Computer Science, Berlin: Springer-Verlag, 2003 [Proceedings of the 4th International Workshop on Conceptual Modeling Approaches for e-Business Dealing with Business Volatility, Chicago, United States, October 2003].
10. Prakash N., Srivastava S. Engineering Methods For Schema Transformation: Application to XML. IFIP TC8/WG8.1 Working Conference on Engineering Information Systems in the Internet Context, September 25-27, 2002, Kanazawa, Japan. Kluwer Academic publishers, pp. 153-175.
11. Rosca D., Greenspan, S., Wild, C.: Enterprise Modeling and Decision-Support for Automating the Business Rules Lifecycle, Automated Software Engineering, Vol. 9, n. 4, pp. 361, 44, 2002.
12. Ross R., Healy K. Organizing Business Plans. The Standard Model for Business Rule Motivation. Business Rules Group. 2000.
13. Ross R. Principles of the Business Rule Approach. Addison Wesley, 2003.
14. Vasilecas O., Smaizys A. Business rule based knowledge base integrated intelligent system framework. Information Sciences 2005, vol. 34, Vilnius. 2005, pp. 195-200.
15. Vasilecas O., Smaizys A. Business rule specification and transformation for rule based data analysis. Proceedings of 19th International Conference on Systems for Automation of Engineering and Research (SAER 2005), Varna, St. Konstantin resort. 2005, pp. 35-40.
16. Vasilecas O., Smaizys A. Business rule based data analysis for decision support and automation. Proceedings of 7th International Conference on Computer Systems and Technologies (CompSysTech 2006), Varna, 2006

A Comparative Review of Approaches for Database Schema Integration[1]

Aiste Aleksandraviciene, Rimantas Butleris

Department of Information Systems, Kaunas University of Technology,
Studentu St 50–308, LT–51368, Kaunas, Lithuania,
{aiste.aleksandraviciene, rimantas.butleris}@ktu.lt

1 Introduction

Any useful information system (IS) needs a conceptual schema, which produces an abstract, global view of all data, managed in the organization. Even though conceptual modeling represents only a small portion of the total information system development effort, its influence on the quality of the final result is probably greater than any other phase. Database (DB) schema is a major determinant of system development costs, system flexibility, integration with other systems and the ability of the system to meet user requirements [13]. This leads to the conclusion that conceptual data-base design is an essential activity of IS development [12].

Integration of local schemata is a popular way to compose correct and complete global database schema – the core of any IS. Schema integration has been a research area since the late 1970s [3] and many approaches to schema integration for various data models have been proposed [7].

This paper gives the results of comparative review performed on six database schema integration methods and models. For this study we used a modification of comparative analysis framework, proposed by Batini et al. [2]. This framework is based on the standpoint that any approach for schema integration is a mixture of four activities. Those are *pre-integration*, *comparison of schemas*, *conforming the schemas*, and finally *merging and restructuring*.

We used relatively old framework to define fundamental criterions applied to compare approaches for schema integration. However, we applied these criterions to new approaches mostly, so it was necessary to add certain new criteria that

[1] This work is supported by Lithuanian State Science and Studies Foundation according to Eureka programme project “IT-Europe” (Reg. No. 3473).

were not mentioned in the survey of Batini at al., i.e., cardinalities of corresponding schema elements and automation levels.

In this paper we focus on two contexts of using the term "schema integration". Those are *conceptual DB schema design* and *distributed DB management*. Although they are different enough, the overall philosophy of schema integration applies to both [14].

The comparative review proposed in this paper was strongly motivated. The results of this review were used to derive a set of desirable features for a special new algorithm suitable for entity-relationship (ER) schema integration in the context of conceptual DB schema design. These features summarize previously reviewed approaches.

In the next section we introduce the basic steps of our approach for conceptual DB schema design, including the above-mentioned algorithm. Six approaches used for the comparative review are introduced in Section 3, and the review itself is presented in Sections 4 and 5. Section 6 gives a set of expected features for ER schema integration algorithm we are going to develop. Conclusions and hints on future research are disclosed in Section 7.

2 Database Schema Design on the Basis of Data Flows

Conceptual DB schema design on the basis of data flows specification is a component of the functional requirements specification method named ODReS (Output-Driven Requirements Specification). Using this method the process of information system development starts with the analysis of requirements for outputs (e.g., reports, summaries) in order to define the limits of problem domain; then the inputs that form required outputs are analyzed. The use of outputs for requirement elicitation reduces the chances of misunderstandings of end-user requirements because the output information is the best for end-user to understand [5].

ODReS method relies on the standpoint that any organization is the sys-tem of incoming and outgoing information flows with links among them. On the basis of this standpoint data flows specification is composed. It is stored in a repository as relational metadata.

There are four steps of conceptual DB schema design on the basis of data flows specification [6]:

1. Specification of the structure of incoming information, referred to as *data resources*. This step produces a number of local schemata, one for each data resource. For conceptual modeling the ER model is used.
2. Specification of the links between data resources. In this step the relationships, reflecting semantic dependencies between elements from different local schemata, are defined. Set of these relationships, showing data transition between two data resources, is referred to as *data flow*.
3. Integration of local schemata, composed during the first step of conceptual DB schema design. In this step data flow structure is used to find out what local schemata can be merged. As data flow shows semantically dependant

elements between two local schemata, it is used as a basis for schema integration while defining corresponding fragments of component schemas. The activity of local schemata integration produces a global DB schema of information system. Generally the process of integration is iterative.
4. Verification of integrated schema using specification of the structure of required outputs.

It is obvious, that the fundamental principle of conceptual DB schema design described above is integration of local ER schemas (each composed for one data resource) using the structure of data flows. The first two phases of this decentralized bottom-up approach were presented in earlier publications [6, 8], and this paper is dedicated for the third one.

3 Approaches for Database Schema Integration

As it has been already mentioned, we applied the comparative review framework to six methods and models used for DB schema integration.

Table 1. Methods and Models of Comparative Review

Author/s	View integration	Database integration	Data model
Adam and Gangopadhyay [1]	+		Conceptual dependency diagram
Bellström [3]	+		Enterprise model
Chao [7]		+	Object-oriented model
Lawrence [11]		+	Entity-relationship model
Navathe et al. [14]	+		Entity-category-relationship model
Sattler et al. [16]		+	Any model

It is evident from Table 1, that three of them are used for view integration, while the other three are applied when integrating databases. View integration is a task of conceptual DB schema design and database integration is a task of distributed DB management.

The basic input to schema integration process is a number of component schemata and a basic output is an integrated schema. Table 2 shows specific inputs and outputs for each schema integration method or model. View integration focuses on integrating all the local schemas into a global conceptual one, while database integration focus on providing a global view of all existing databases [3].

Table 2. Inputs and Outputs*

Author/s	Inputs	Outputs
Adam and Gangopadhyay [1]	*n* database views	Global DB schema, transformed from global conceptual diagram
Bellström [3]	User-defined views	Conceptual DB schema
Chao [7]	Two component schemas	Integrated schema
Lawrence [11]	*n* local schemas of existing databases Relational semantic metadata Global dictionary	Global view *m* projections of global view for each user-group Global to local mappings
Navathe et al. [14]	*n* user views	Global DB schema Global to local mappings
Sattler et al. [16]	Local schemas of heterogeneous databases Global object types	Global view of DB federation Global to local mappings

* This table was filled using original terminology given by authors.

4 Process of Database Schema Integration

Any integration methodology is a mixture of the following activities [2]:

1. *Pre-integration*. An analysis of local schemata is carried out. The analysis made governs the choice of schemas to be integrated and the order of integration. A number of schemas to be integrated at a time and the amount of designer interaction are also decided in this step.
2. *Comparison of the schemas*. Schemas chosen for integration are compared to determine the correspondences among concepts and detect possible conflicts.
3. *Conforming the schemas*. In order to integrate schemas, the detected conflicts between them have to be resolved so that it was possible to merge them.
4. *Merging (a) and restructuring (b)*. Schemas are merged in this step. The result of final or intermediate schema merging is analyzed and, if necessary, restructured in order to achieve several desirable qualities.

Table 3 shows how the steps of each approach cover these activities.

Table 3. Schema Integration Activities

Author/s	Step 1		Step 2		Step 3		Step 4a	Step 4b
Adam and Gangopadhyay [1]	x	⟶	x	⟶	x	⟶	x	—

Bellström [3]	–		x	→	x	→	x	→	x
Chao [7]	–		x	→	x	→	x		–
Lawrence [11]	x	→	x	→	x	→	x	→	x
Navathe et al. [14]	x	→	x	→	x	→	x	→	x
Sattler et al. [15]	x	→	x	→	x	→	x		–

Note, that some approaches allow a feedback between two steps. Chao [7] provides a global loop from the end of the process to the initial comparison activity, meaning that there can be some correspondences that cannot be specified until the integrated schema is being constructed.

Navathe et al. [14] and Sattler et al. [16] provide an iterative execution of comparison and conforming steps before any merging is performed.

4.1 Pre-integration

In this step the decision on strategy for integration processing is made.

Strategies for schema integration processing are classified into *binary* and *n-ary* [2, 4].

Binary strategies allow the integration of two schemas at a time (n = 2). They are called *ladder* strategies, when a new component schema is integrated with an existing intermediate result at each step. A binary strategy is *balanced*, when the schemas are divided into pairs at the start and then are integrated in a symmetric fashion [2].

N-ary strategies allow integration of n schemas at a time (n > 2). An n-ary strategy is *one-shot* when n schemas are integrated in a single step, and it is *iterative* otherwise.

Table 4. Strategies for Integration Processing

Author/s	Strategy
Adam and Gangopadhyay [1]	Iterative n-ary
Bellström [3]	Binary, N/A
Chao [7]	Binary, ladder
Lawrence [11]	Binary, ladder
Navathe et al. [14]	One-shot n-ary
Sattler et al. [16]	Binary, N/A

As Table 4 shows, most of the analyzed approaches adopt binary strate-gies. As the complexity of integration increases with respect to the number of schemas integrated at a time, binary strategies simplify the activities of schema comparison and conforming. However, any binary schema integra-tion method typically requires more restructuring then any n-ary schema integration approach [1].

4.2 Comparison of Schemas

As is it evident from Table 3, all approaches of the comparative review provide the activity of schema comparison. Schema comparison is necessary for identifying corresponding elements and detecting conflicts.

A *conflict* between two representations of the same real-world object is every situation that gives rise to these representations not being identical.

Causes for Database Schema Diversity

There are three main causes for DB schema diversity:

1. The same real-world object can be seen from different levels of abstraction, or represented using different properties [10].
2. The same reality can be modeled using different combinations of data model constructs.
3. Incompatible design specifications (improper choices regarding names, types, integrity constraints, etc.).

Cardinality Cases of Corresponding Schema Elements

Suppose we have to compare two local schemas S_1 and S_2. Any element in S_1 can have one or more corresponding elements in S_2. Thus, we can use familiar relationship cardinalities to express four possible cardinality cases between corresponding elements of two different schemas. Those are exact case 1:1 and setoriented cases 1:N, N:1, and M:N [15]. Exact cardinality means that some element in S_1 corresponds to only one element in S_2. If there is a difference in representation of these elements (e.g. price in dol-lars vs. price in euros), a conflict occurs. Set-oriented cardinality 1:N means that some element in S_1 (e.g., attribute "Name") corresponds to multiple elements (e.g., attributes "FirstName" and "LastName") in S_2. There is always a conflict in such a situation. The same situation is in case of cardinality N:1. As cardinality case M:N is a combination of 1:N and N:1, it is the most sophisticated one.

Table 5. Cardinalities of Correspondences (at the element level)

Author/s	1:1	N:1	1:N	M:N
Adam and Gangopadhyay [1]	+			
Bellström [3]	+			
Chao [7]	+	+	+	+
Lawrence [11]	+	+	+	N/A
Navathe et al. [14]	+			
Sattler et al. [16]	+	+	+	

As it is evident from Table 5, all the analyzed approaches support the identification of correspondences with exact cardinalities, and only Chao [7] method can be used to define the full set of cardinality cases.

Types of Conflicts

As it was mentioned previously, various types of conflicts can occur among corresponding elements contained in two or more different sche-mas.

We identified three common types of conflicts: *naming*, *semantic*, and *structural*. Naming conflicts include homonyms and synonyms, while structural conflicts mainly deal with incompatible both schema- and data-level types, key equivalence and dependency conflicts. Semantic conflicts deal with semantic dependencies among two or more schema elements, expressed using modeling constructs of the same type (e.g., entities, classes, relations, attributes, etc.). A situation of exact semantic dependency is referred to as *equivalence* ($A \equiv B$), and it is not a conflict. But there are three situations of inexact semantic dependencies. Thus, there are three kinds of semantic conflicts: *containment* ($A \subseteq B$), *overlapping* ($A \cap B$), and *disjointness* ($A \cap B = \varnothing$). All of them can be detected using any of the reviewed approaches except the method, proposed by Bellström [3]; he does not mention semantic conflicts in his paper.

Table 6 shows, what kinds of naming and structural conflicts can be detected using reviewed methods and models.

Table 6. Naming and Structural Conflicts

Author/s	Naming	Structural
Adam and Gangopadhyay [1]	Homonyms Synonyms	–
Bellström [3]	Homonyms Synonyms	Dependency conflicts Key conflicts
Chao [7]	Homonyms Synonyms	Domain type conflicts Object type conflicts (attribute-class)
Lawrence [11]	Homonyms Synonyms	Data level type conflicts Dependency conflicts Schema level type conflicts (attribute-entity, attribute-relationship, and entity-relationship)
Navathe et al. [14]	Homonyms Synonyms	Dependency conflicts (differences in roles, degree, and cardinalities) Key conflicts
Sattler et al. [16]	Homonyms Synonyms	Attribute conflicts Key (equivalence) conflicts Meta conflicts Table structure conflicts (overlapping fields)

4.3 Conforming the Schemas

Merging of schemas is possible, only if naming and structural conflicts are resolved. This is the essential condition for schema integration.

There is a plenty of suggestions, how to resolve naming conflicts. How-ever, we structural conflicts still remain an open issue. As Table 7 shows, most authors tend to agree with resolving homonym and synonym con-flicts by applying "rename" operations. Giving different names or prefix-ing eliminates homonyms. Unifying names eliminates synonyms. In some specific cases, if data model allows, synonyms can be excluded avoiding "rename" operations, e.g., in enterprise model a mutual inheritance is used.

Resolution technique used to resolve key conflicts is twofold: Bell-ström [3] recommends including both keys into the integrated schema, while Sattler et al. [16] suggest adopting only one of them.

Problems with incompatible schema level types are resolved by conver-sion from one type into the other, i.e., "upgrade" [7] and "convert" [11, 14] operations. Data level type conflicts are resolved using various conversion functions [16].

There is a whole methodology devoted mainly for resolving dependency conflicts in [14] and a newsworthy suggestion of Bellström [3] to adopt semantically weaker relationship in such case.

4.4 Merging and Restructuring

After the merging and restructuring is performed, the integration of schemas is completed and integrated schema is produced.

Such operations as "merge" [1], "create" [7, 14], and "integrate" [11] are applied in order to merge the same objects. Semantic conflicts are re-solved by using generalizations and/or specializations. Dissimilar objects are simply added to the integrated schema.

Integrated schema is analyzed and, if necessary, restructured to achieve desirable qualitative criterions [2]. Restructuring the integrated schema means eliminating redundancies, adding new components, etc.

Table 7. Operations for Schema Transformations

Author/s	Conforming	Merging and restructuring
Adam and Gangopadhyay [1]	Rename	Add (dissimilar components) Generalize/specialize Merge
Bellström [3]	Prefix Adopt semantic weaker dependency	Create inheritance hierarchy Include both keys Set mutual inheritance
Chao [8]	Coerce Concatenate Rename	Compose Combine Create

	Upgrade	Create inheritance hierarchy Generalize Specialize
Lawrence [11]	Convert to entity Convert to relationship Standardize names	Add Change Integrate
Navathe et al. [14]	Convert entity to category Rename	Add Create Generalize Remove redundant relationships Specialize
Sattler et al. [16]	Create a mapping table Transpose Use conversion functions	Apply “Join” operator Apply “Union” operator

5 Schema Integration Process: Manual or Automatic?

The automation level measures the ability of an approach to be automated. In general, there can be three levels of automation: *manual*, *semi-automatic*, and *automatic*. We examined the main activities of each approach against these criteria (see Table 8).

Table 8. Automation Levels

Author/s	Comparison	Integration	Reasoning
Adam and Gangopadhyay [1]	Semi-automatic	Semi-automatic	Integration rules (on the basis of Conceptual Dependency Theory).
Bellström [3]	Manual	Manual	Heuristic algorithms.
Chao [7]	Semi-automatic	Automatic (limited support)	– Correspondence assertions in form of predicates. – 5 integration rules: algorithmic steps containing primitive algebraic operators.
Lawrence [11]	Automatic	Automatic (limited support)	– Global dictionary (contains unique names of concepts represented in a global view). – RIM specifications (relational semantic metadata – a kind of export schema). – Mapping rules.
Navathe et al. [14]	Semi-automatic	Semi-automatic	– Hierarchal schema for relationship comparison against three criteria. – Integration rules.
Sattler et al. [16]	Automatic	Semi-automatic	– Interactive conflict detection and resolution, based on examples.

	– Advanced mechanisms for conflict resolution, supported by $F_{RA}QL$.

Referring to Table 8, it is evident, that none of the approaches provide fully automated both schema comparison and integration process.

6 Expected Features for a Future Approach

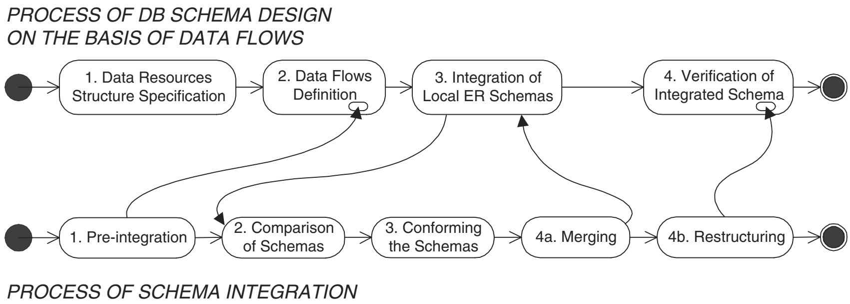


Fig. 1. Links between the activities of two processes

Fig. 1 shows, how the activities of DB schema design, introduced in Sect. 2, are related to the basic activities of schema integration process. Definition of data flows includes pre-integration as sub-activity, because data flows, when defined, give the recommendations on grouping and sequencing schemas for integration. Integration of local ER schemas includes the activities of comparison, conforming, and merging the schemas. Finally the integrated schema is verified to check, if it is able to form all required outputs. If verification fails, restructuring is necessary.

Expected features for the future approach are given in Table 9. Note, that possible types of conflicts are determined by the boundaries of relational metadata structure.

Table 9. A Set of Expected Features for the Future Approach

Feature	Description
Context	Conceptual DB schema design (view integration)
Data model	ER model, Information Engineering (IE) notation [9]
Inputs	– *n* local schemata (one for each data resource) – *m* specifications of data flows structure – Data flows specification (relational metadata)
Output	Conceptual DB schema
Strategy	Binary, both ladder and balanced
Conflicts:	
Naming	Synonyms
Semantic	Containment, overlapping, disjointness

Structural	– Attribute-entity conflicts – Key equivalence and dependency conflicts

7 Conclusions

This survey is expected to be useful both to developers of new approaches for DB schema integration and to users who need to select from a variety of approaches. For the future research we are going to extend our survey with the analysis of ontology-based schema integration approaches. And we also plan to develop a semi-automated tool for bottom-up DB schema design, including ER schema integration.

References

1. Adam NR, Gangopadhyay A (1995) An N-ary View Integration Method Using Conceptual Dependencies. In: Proc. of 28th Hawaii Intl. Conf. on System Sciences. IEEE Computer Society, Washington, pp 391–397
2. Batini C, Lanzerini M, Navathe SB (1986) A Comparative Analysis of Methodologies for Database Schema Integration. ACM Computing Surveys 18(4):323–363
3. Bellström P (2004) Using Enterprise Modeling for Identification and Resolution of Homonym Conflicts in View Integration. In: Vasilecas O (ed) Information Systems Development: Advances in Theory, Practice and Education. Springer, pp 266-276
4. Boman B, Bubenko J, Johanesson P, Wangler B (1993) Models, Concepts, and Information: An Introduction to Conceptual Modelling for Information Systems Development. Royal Institute of Technology, Stockholm
5. Butkiene R, Butleris R, Danikauskas T (2002) The Approach of Consistency Checking of Functional Requirements Specification. In: Proc. of 6th World Multiconference on Systemics, Cybernetics and Informatics, vol XVIII. Orlando, pp 67–72
6. Butleris R, Danikauskas T (2004) Conceptual Data Model Using Functional Requirements Specification Method. In: Proc. of 16th Conference on Advanced Information Systems Engineering (CAiSE), vol 1. Riga Technical University, Riga, pp 221–232
7. Chao Ch-M (2001) Schema Integration between Object-Oriented Databases. Tamkang. Journal of Science and Engineering 4(1):37–44
8. Danikauskas T, Butleris T, Drąsutis S (2005) Graphical User Interface Development on the Basis of Data Flows Specification. In: Proc of 20th International Symposium on Computer and Information Sciences, LNCS 3733. Springe-Verlag, Berlin Heidelberg, pp 904-914

9. Halpin T (2002) Metaschemas for ER, ORM and UML Data Models: A Comparison. Journal of Database Management 13(2):20–29
10. Johannesson P (1993) Schema Integration, Schema Translation and Interoperability in Federated Information Systems. Ph.D. thesis, Stockholm University
11. Lawrence R (2001) Automatic Conflict Resolution to Integrate Relational Schema. Ph.D. thesis, University of Manitoba
12. Olivé A (2005) Conceptual Schema-Centric Development: A Grand Challenge for Information Systems Research. In: Pastor O, Cunha JF (eds) CAiSE '2005, LNCS 3520. Springer-Verlag, Berlin Heidelberg, pp 1–15
13. Moody DL, Shanks GG (2003) Improving the quality of data models: empirical validation of a quality management framework. Information systems 28:619–660
14. Navathe SB, Sashidhar T, Elmasri R (1984) Relationship Merging in Schema Integration. In: Proc. of 10th Intl. Conf. on VLDB. Singapore, pp 78–90
15. Rahm E, Bernstein PhA (2001) A survey of approaches to automatic schema matching. The VLDB Journal 10:334–350
16. Sattler KU, Conrad S, Saake G (2003) Interactive example-driven integration and reconciliation for accessing database federations. Information systems 28:393–414

Strategic Use of Customer Relationship Management (CRM) in Sports: The Rosenborg Case

Bjørn Furuholt*, Nils Georg Skutle**

* School of Management, Agder University College, Kristiansand, Norway
E-mail: Bjorn.Furuholt@hia.no
** Buysec AS, Kongsberg, Norway

Abstract: Today's sports industry has turned into a billion industry on the same level as more traditional industries. Present-day information technology and Customer Relationship Management (CRM) has enabled companies to deal with vast numbers of customers and to establish one-to-one communication with each of them. Lately, sports clubs have also discovered the benefits of introducing CRM. A sports club's most important customers are its supporters. They are different from traditional customers in that they are characterized by a strong sense of loyalty to their favourite club. This article shows how one sports club, the Norwegian football club Rosenborg Ballklub (RBK), can use CRM as a strategic tool. Due to the limited material in this subject area, this is an explorative piece of work, based on examination of literature, in-depth interviews at RBK and collection of questionnaires from a selection of European football clubs.Our findings indicate that there are many similarities between a football club and a traditional company when CRM is introduced. In addition, for clubs, the CRM system must be adapted to the supporter role, and we propose the introduction of the concept Supporter Relationship Management in order to take better care of the strong ties between a club and its supporters.

1 Introduction

Professional team sports have become a billion industry where the biggest clubs in European top football have earnings that surpass €200 million [3]. This has led to the fact that the biggest teams operate more like large companies than traditional sports clubs and this increased professionalism has gradually spread to the smaller clubs across Europe.

A professional sports club's most important customers are its supporters. They provide direct income through buying tickets to matches and supporter gear as well as indirect earnings through sponsor and TV agreements, which are closely linked to how popular each club is. Gradually the clubs have entered "the digital world", which has paved the way for new channels to reach their supporters. and strengthen their relations with each individual person. This is what we call Customer Relationship Management (CRM).

There are few areas in information systems that have attracted as much attention as CRM in recent years. When it comes to the combination of CRM and sports, there are few articles with an academic starting point.

Rosenborg Ballklub (RBK) is Norway's biggest sports club regardless of sports type, when it comes to both success on the football field and financially. RBK started a CRM project for the 2005 season as the first sports club in Norway. Our research question is connected to this project and is formulated as follows: "How do we proceed to implement a successful CRM solution at RBK, and how can CRM be used strategically?" Primarily we will concentrate on sports specific circumstances, the use of CRM in relation to the club's supporters. When it comes to using CRM in relation to other customers, the clubs are not very different from traditional companies. In order to answer the question we have examined literature concerning general CRM implementations, carried out an investigation at a selection of European football clubs, and finally examined the situation in Norway and specific RBK conditions.

The paper is organized as follows. This introduction is followed by a review of relevant literature based on three main topics: sports and football economics, supporter loyalty, and CRM. Chapter three presents the methodology, whereas chapter four includes a survey of the findings from the questionnaire and the in-depth interviews at RBK. The findings are discussed in chapter five, and the article is rounded up with conclusions in chapter six.

2 Literature review

2.1 Football economics

Sports have evolved from a leisure time activity for amateurs in the beginning of the 20th century to the billion industry of today. This has resulted in the fact that clubs nowadays must to a great extent regard themselves as companies that operate in a competitive market. However, there are basic differences between the commercial mechanisms in professional team sports and traditional business activity. A company is successful if it can get rid of all competition and secure a monopoly situation. In sports, on the other hand, it would not be favourable for a team to reduce competition [4]. The reason is that added value in professional sports is inherently a joint achievement by the different teams in the league. Neale

[9] illustrates this by what he calls “the Louis Schmelling paradox”. World champion Joe Louis’ earnings were higher if there was an evenly matched contender available for him to fight than if the nearest contender was relatively weak. The same principle applies to professional team sports. This paradox goes against all logic in traditional economics where you strive for monopoly in order to eliminate the competitors. Neale [9] addresses this paradox by distinguishing between ‘sporting’ and ‘economic’ competition. Sporting competition is more profitable than sporting monopoly. When it comes to economic competition the different teams co-operate in order to create a joint product.

2.2 Supporter loyalty

Sports clubs are in a favourable position because their supporters are very loyal compared with traditional customers [13,16]. Loyalty can be understood as a two-dimensional concept with a psychological and behavioural dimension, and Backman and Cromton [1] developed a loyalty model, which takes these two dimensions, into account. Based on Backman and Crompton’s model Mahony et al. [8] developed a scale in order to measure the psychological commitment among supporters. This scale was used to place the supporters according to the four categories in Backman and Crompton’s loyalty model: supporters with high loyalty, spurious loyalty, latent loyalty, and finally low or no loyalty.

Guilianotti [6] has made a taxonomy in which he, just like Mahoney, identifies four categories with ideal supporter types: supporter, fan, follower and flâneur. He employs two dimensions to differentiate between the different types: traditional/consumer dimension and hot/cool dimension that will say something about how strongly a supporter identifies with the club. If we compare Guilianotti’s ideal types directly with Backman and Crompton’s loyalty model, we learn that it does not quite add up. For instance we see that the supporter, who would be the ideal type with the strongest sense of loyalty to the club, end up in the category of latent loyalty. The fan, on the other hand, ends up in the category of high/genuine loyalty. Of this reason we will work out our own adjustment of these models, which will better demonstrate how the ideal types can fit in with several loyalty categories (figure 1).

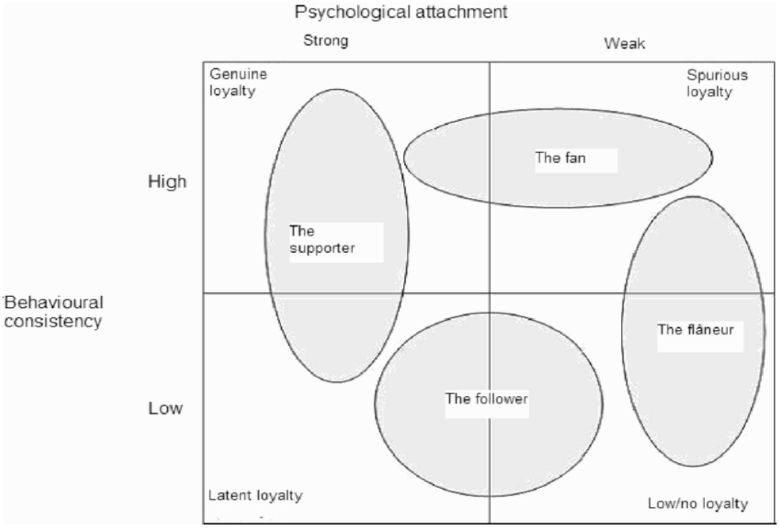


Fig. 1. The relation between ideal types and the loyalty model

This turns out to be more correct than a direct transfer because the supporter base in professional football clubs is consequently not a homogeneous group, but is made up of different types with individual motivations for supporting the club [6, 8, 11].

2.3 CRM

The concept of Customer Management has been around for as long as people have been trading goods and services and as the population has increased, it has become more and more difficult to keep track of customers [5]. Today's companies introduce Customer Relationship Management in order to make up for this. Zablah et al. [19, p.477] have gone through CRM literature and identified five different perspectives on CRM: process, strategy, philosophy, ability and technology. Based on this, they make an attempt at reaching a joint conceptualization of CRM. They consider the process to be the most fruitful perspective and arrive at the following definition of CRM:

"CRM is an ongoing process that involves the development and leveraging of marked intelligence for the purpose of building and maintaining a profit-maximising portfolio of customer relationships" [19, p. 480].

Unfortunately, a lot of CRM projects have failed and have by far kept their promise. The Gartner Group have found that 70% of all CRM projects result in loss or no improvement in the company's financial performance (stated in [12]). A series of articles discuss critical success factors and pitfalls that must be avoided in connection with CRM [e.g., 7, 10, 14, 15]. In these articles there are four areas that stand out as the most important areas: top management support, strategic thinking, organisational changes and management of these, and finally the role of technology in CRM.

CRM's strategic nature is an important argument for necessary top management support. The most important change the organisation must go through is to focus on the customer. Strong and sound change management is absolutely necessary if the organisational changes are to succeed. In investigations carried out in this area it turned out that lack of adequate change management was stated as the main reason that CRM projects failed in 87% of all the cases [7, 14].

The final area that needs special attention in a CRM project is technology. The biggest mistake companies make when it comes to technology is to place too much importance on it, and to think that a CRM solution will handle all customer relations and automatically make the organisation customer-oriented.

There has been little mention in academic literature of the use of CRM in the world of sports. It seems, however, to be a consensus that CRM and sports belong together [2, 17, 18]. There is no indication that success factors such as top management support, change management and strategic thinking become more or less important because we are talking about sports. What makes a customer at a sports club different from a traditional customer is that he considers himself as a supporter and not a customer. And, this important difference is what separates CRM in the sports industry from traditional CRM.

3 Methodology and data collection

Since CRM in football clubs is a relatively unexplored area, this article is exploratory by nature. RBK was chosen as a special case for further studies because it is the first Norwegian club to introduce CRM. The interviews made at RBK focused on three different aspects: What special circumstances are there in a football club? What special circumstances are there in a Norwegian football club? How can CRM fit in? Three semi-structured interviews were conducted, with the managing director, the marketing director and a project manager in the marketing department. The interviews were taped and then transcribed.

In addition a questionnaire, in English, was sent to 13 different clubs in Europe. These varied in size from top clubs such as AC Milan, to clubs of the same size as RBK such the Danish FC Copenhagen. The purpose of the questionnaire was to find out what kind of practice the various clubs had in relation to CRM and CRM strategy. By studying similarities between the clubs we wanted to bring up "best practices" in CRM usage in football clubs. The questionnaires were sent along with a cover letter from RBK, four were sent to the club's official address and nine were addressed to a contact person in the club, picked out together with RBK. We received six answers, all among the nine that were sent to specific contact persons. All contact persons were part of the clubs' top management. The questionnaires were completed by the respondents.

4 Findings

4.1 The clubs' responses to the questionnaire

We have categorized the six clubs, which returned the questionnaire, in three categories: top clubs (AC Milan, Juventus and Bayern München), intermediate clubs (Borussia Dortmund and PSV) and Scandinavian clubs (Djurgårdens IF from Sweden).

Four clubs, Bayern München, AC Milan, Borussia Dortmund and Juventus, stated that they had a CRM strategy. A common theme among these clubs is that they will develop stronger relationships with their supporters to get more satisfied customers. PSV is the only club that does not state any form of strategic thinking behind its CRM initiative. They consider CRM as a technical solution only.

The size of the CRM systems in the clubs varies from a simple web based solution at Djurgården to CRM as an integrated part of a large-scale Enterprise Resource Planning (ERP) system at Borussia Dortmund and Bayern München. Both clubs have systems that are specially adapted to the needs of a football club. In order to achieve the clubs' goals concerning customer satisfaction, the employees have access to updated customer information from several sources such as membership records, booking systems and online sales of supporter gear.

Based on the answers we can draw the conclusion that it is important for a CRM system to process customer data collected from several sources and put them together in an overall presentation of the customer. It is also clear from the answers that the system should be specially adapted for use in a football club.

AC Milan and Bayern München have run their CRM systems for a few years, and are a little ahead of the other clubs. These two clubs recognize the importance of the continuous development of both strategy and software in order to meet new challenges and possibilities.

Three points are mentioned when the clubs bring up advantages of CRM: improved customer information, efficiency and improved customer relationship. Our conclusion is that the most important advantage of CRM is improved customer data. When customer data are easily available, the clubs could increase the efficiency of their processes, both internally and towards customers.

Another important CRM system characteristic mentioned is the fact that it makes it possible for the clubs to tailor-make offers for the various customer segments. The number of categories varies from two at Juventus, to six at Bayern München and Borussia Dortmund and there are four main categories that stand out: ticket buyers, supporter gear buyers, sponsors and members.

It seems like none of the clubs have a carefully thought through a strategy for how to use CRM to get new supporters. CRM is first and foremost used to follow up existing supporters and for this they will primarily employ two methods. The active method is to use the CRM system to tailor information and offers and send it to the supporters by way of various information channels. In the passive variant, CRM is used to make personalized membership sites on the club's home page.

Some clubs, such as AC Milan and Bayern München appear to make use of both variants.

4.2 In-depth Rosenborg interviews

CRM can serve as a support for both the club's core product and the various by-products, and RBK hopes that CRM can contribute to improve customer communications and to coordinate marketing activities across customer groups.

Supporters are customers who are often very emotionally attached to their favourite club. This attachment to the club is clearly an advantage in a marketing perspective, but it is also a disadvantage. When supporters give so much to their club, they also expect that the club repays these affections at one level or the other. If supporters feel they are treated like ordinary customers, it may result in the opposite reaction of the desired effect.

European clubs operate in a much bigger market than RBK. In a CRM context this should not have any direct impact on neither the implementation nor the development of a strategy. It will, however, have an impact on the risk involved in such a project. As in the words of the managing director:

> *"... we are a lot more careful when investing, whereas Barcelona or United can with a greater degree of certainty get some return on the investment because it does not have to be that great. It is just that the number they serve is so much larger."*

In other words, it seems like the risk associated with these kinds of projects is in inverse ratio to the size of the market.

Tradition is another difference between Norwegian and European clubs. All the while the Norwegian clubs based their economy on selling hot dogs or coffee during the break, the major European clubs have handled ten thousands of spectators for many years. This lack of professionalism in the organisation, may present a problem to the Norwegian clubs when they are about to introduce CRM. As for RBK this means that they must attach great importance to training as an important part of the investments. It appears from the interviews that RBK considers organisational factors as the greatest impediments to a successful CRM project:

> *"We are very old-fashioned when it comes to our work routines, and I believe this is one of the major challenges when it comes to getting full benefits from a CRM solution."* – (Marketing manager.)

The challenge is to get the organisation to not only use the new technology, but get an understanding of its utility value as well. This applies to both the top management and the employees, and it underlines the assertion that RBK must focus on training as part of its CRM project.

5 Discussion

Based on the literature review we got four critical success factors to fully benefit from CRM: top management support, strategic thinking, change management and technology. In addition to these four important areas, a CRM project should also be an evolutionary project that is constantly under development. This implies that a CRM project does not necessarily have to involve the whole organisation at the same time. By choosing a smaller and focused CRM project the chance of success will be better [15].

Top management support triggers the whole process. When the CRM strategy falls into place, a change process in the organisation is started.

When it comes to technology, there is a different practice among the clubs, but none of the clubs in question, had systems that were delivered straight from the supplier's shelves, without any adjustments. In other words, it is not only the technology that affects the club, but the club affects the technology as well. This adjustment is not done once and for all, but it is a continuous improvement process.

RBK is in the first phase of its CRM project. They have not yet implemented a full CRM solution. They recognized the need for organizational changes, and they needed to increase the level of IT knowledge in the organisation. Therefore, in the case of RBK, training should be added as a fifth success factor. Risk was another point of interest to RBK and they must be certain of the project's profitability before the project starts up and thus a business case should be presented.

Based on the previously discussed success factors and the additional RBK factors (business case and training), we arrive at an implementation process model as presented in figure 2.

It is alleged that it is a waste of time and resources if you get started on a CRM project, which is not vital to the company's competitiveness [15]. RBK's core product is a competitive football team. CRM can not make Rosenborg play better football, but it may support the by-products and thereby give the club an improved financial foundation. We believe that RBK, in addition to CRM must introduce Supporter Relationship Management (SRM). SRM will be an integrated part of the club's CRM system and use the same underlying functionality. Whereas CRM takes care of the club's traditional customer relationships, the main function of SRM will be to take care of the club's relationship with the supporters.

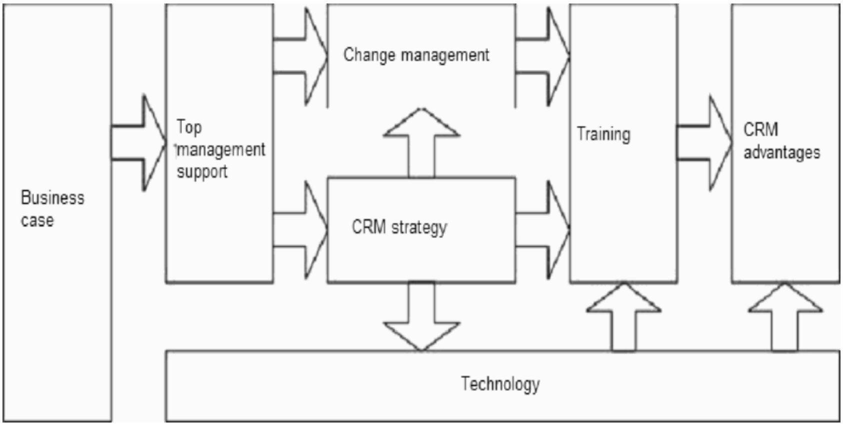


Fig. 2. CRM implementation process in a sports club

The results from the questionnaires demonstrate with total clarity that the supporter is the club's most important customer. One of the great advantages of SRM is that it makes it possible to have one-to-one communication with the supporter. RBK can differentiate between them based on their loyalty, and tailor marketing measures on the basis of the position of each supporter in the loyalty model, like newsletters and designated areas at the club's home page.

Richardson and O'Dwyer [13] found out that if supporters change loyalty, it occurs mainly among young people under the age of 10. This means that if a club is able to capture a young person's interest before someone else, it will most likely keep this supporter for the rest of his life. AC Milan has linked its CRM system to their activities that are aimed at young people because they recognize the importance of establishing a relationship with their supporters at an early stage.

In some cases the supporters are traditional customers for the club. They buy tickets and supporter gear. Some supporters may, however, spend very little money on the club. A traditional company with a traditional CRM system would spend little time and efforts on customers like that. However, a great deal of the club's market value is based on the outside world's conception of the size of the club's body of followers. Consequently, a SRM system must take these supporters into consideration as well and to keep them in the system as a reference for potential sponsors.

6 Conclusion

In this article we have shown that CRM can be used strategically in a football club. Based on the findings in the questionnaire responses we learn that the biggest clubs have a CRM strategy and that CRM is a prioritized area to these clubs. The two greatest advantages, which are stated by the clubs, are improved customer relationships and increased efficiency of internal processes. In order to consider

CRM as a strategic tool and to develop a CRM strategy you are dependent on top management support which is vital to a strategic foundation for CRM, and also important when carrying out the necessary organisational changes. The CRM project should also be concentrated only on areas that are crucial to the club's ability to compete. For football clubs this means that the chief concern is to focus on their supporters.

In addition, we point out how the supporters are different from traditional customers. This is something a CRM system in a football club must take into account. Guilianottis [6] ideal types of supporters combined with a survey of the club's different customers can be applied to segment supporters in a useful way, and we suggest that the Supporter Relationship Management (SRM) concept is introduced as well.

When it comes to RBK, two factors emerged, which were important in relation to CRM. One factor was the need to present a business case to map the risk involved in a CRM project, the other was the importance of training in an organisation that does not have the required level of IT competence. Rosenborg can make use of the findings in this paper in the planning of their continued CRM efforts. The most important findings are a survey of the critical success factors in a CRM implementation, a CRM implementation model (figure 2), and tools linked to the SRM concept.

This is an explorative piece of work, and the article is based on a limited selection of data. We therefore envisage several opportunities for further research, both through expanding the data basis with more clubs from more countries, and through following a specific project more closely over a longer period of time. In particular this can give us a better understanding of the implications of SRM in detail.

References

1. Backman SJ, Crompton JL (1991) Differentiating Between High, Spurious, Latent, and Low Loyalty Participants in Two Leisure Activities. Journal of Park and Recreation Administration 9(2)
2. Beal B (2004) CRM – it's a brand new ball game. SearchCRM.com, Available online at: http://searchcrm.techtarget.com/originalContent/0,289142,sid11_gci949574,00.html (Accessed on 03 May 2005)
3. Deloitte (2005) Football Money League: The Climbers And The Sliders. February 2005
4. Dobson S, Goddard J (2001) The Economics of Football. Cambridge University Press, Cambridge, UK
5. Goodhue DL, Wixom BH, Watson HJ (2002) Realizing Business Benefits Through CRM: Hitting the Right Target in the Right Way. MIS Quarterly Executive 1(2): 79-94

6. Guilianotti R (2002) Supporters, Followers, Fans and Flaneurs: A Taxonomy of Spectator Identities in Football. Journal of Sport & Social Issues 26(1): 25-46
7. Kale SH (2004) CRM Failure and the Seven Deadly Sins. Marketing Management 13(5): 42-46
8. Mahony DF, Madrigal R, Howard D (2000) Using the Psychological Commitment to Team (PCT) Scale to Segment Sport Consumers Based on Loyalty. Sport Marketing Quarterly 9(1): 15-25
9. Neale WC (1964) The peculiar economics of professional sports. Quarterly Journal of Economics 78: 1-14
10. Petersen GS (2004) Best Practices and Customer Relationship Management (CRM). Business Credit, January 2004: 48-49
11. Quick S (2000) Contemporary Sport Consumers: Some Implications of Linking Fan Typology With Key Spectator Variables. Sport Marketing Quarterly 9(3): 149-156
12. Reinartz WJ, Krafft M, Hoyer WD (2004) The Customer Relationship Management Process: Its Measurement and Impact on Performance. Journal of Marketing Research 41: 293-305
13. Richardson B, O'Dwyer E (2003) Football supporters and football team brands: A study in consumer brand loyalty. Irish Marketing Review 16(1): 43-53
14. Rigby DK, Reichheld FF, Schefter P (2002) Avoid the Four Perils of CRM. Harvard Business Review 80(2): 101-109
15. Rigby DK, Ledingham D (2004) CRM Done Right. Harvard Business Review 82(11): 118-129
16. Shank MD (2005) Sports Marketing. A Strategic Perspective. Prentice Hall, USA
17. Waltner C (2000) CRM: The New Game In Town For Professional Sports. informationweek.com. Available online at: http://www.informationweek.com/801/crm.htm (Accesssed on 03 May 2005)
18. Weinberger J (2004) Customers for Life. How to transform loyalty into commitment into revenue. CRM Magazine July 2004: 32-38
19. Zablah AR, Bellenger DN, Johnston WJ (2004) An evaluation of divergent perspectives on customer relationship management: Towards a common understanding of an emerging phenomenon. Industrial Marketing Management 33: 475-489

Towards Knowledge Management Oriented Information System: Supporting Research Activities at the Technical University

Marite Kirikova and Janis Grundspenkis

Department of Systems Theory and Design, Riga Technical University, 1 Kalku, Riga, LV-1658, Latvia; e-mail: marite.kirikova@cs.rtu.lv and janis.grundspenkis@cs.rtu.lv

1 Introduction

Recently in knowledge management and business process management communities attention has been focused at knowledge intensive processes (e.g., Verhooef and Qureshi 2005, Papavassiliou et al. 2003, Fedel and Tanniru 2005). There is no doubt that research is a knowledge intensive process as well. However, information systems (IS) support of the research work mainly addresses only two dimensions of scientific activities, namely (1) scientific data retrieval, processing and visualisation and (2) collaboration among scientists (Lincke et al. 1998, Yao 2003, Ludacher et al. 2005, Zhao et al. 2005). Other possibilities to support scientific work, e.g., university level knowledge management of research activities are less investigated and implemented (Hornbostel 2006).

Research activities in the university context should be considered not only from the point of view of generated knowledge and its value for science and industry, but also from the point of view of the value of this knowledge in education processes and attracting financial resources for university needs. In this paper we analyse research activities of the technical university from the process perspective using two process frameworks: (1) American Productivity and Quality Center (APQC) business process framework (APQC 2005) and (2) knowledge value chain (KVC) framework by Holsapple and Jones (2004, 2005). The purpose of the paper is to describe relationships between knowledge processes and business processes of research in terms of both functional and administrative dimensions, as well as to propose guidelines for the development of knowledge management based IS for supporting research activities in the university setting.

The context of university research activities is described in Section 2. Further, in Section 3, the related work in IS support for research activities is considered. In section 4 university research activities are described from the process perspective in the context of APQC and KVC frameworks. In section 5 the guidelines for IS support of research activities at the technical university are proposed. Section 6 consists of brief conclusions.

2 The context of research activities

According to Free Dictionary of Farlex, research is a primary activity of science, a combination of theory and experimentation directed towards finding scientific explanations of phenomena. It is commonly classified in two types: pure research [or basic research], involving theories with little apparent relevance to human concerns; and applied research, concerned with finding solutions to problems of social or commercial importance. The main difference between scientific activities of the technical university and institutions dealing with natural and social sciences is in tangibility of professional knowledge that impacts research activities (standards, certified methodologies, patents, etc.). On the other hand in many technical fields industry is a metrics oriented judge (Geisler 2002) of knowledge provided by the university, which is expected to comprise well integrated scientific and professional skills. The context of research activities at the technical university is illustrated in Fig. 1.

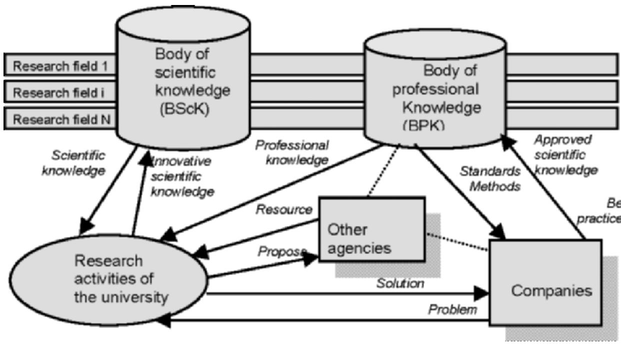


Fig. 1. A simplified scheme of the university research activieties context

As it is shown in Fig. 1, the research activities directly contribute to the body of scientific knowledge (BScK) and indirectly to the body of professional knowledge (BPK). Actually, each field of research has its own scientific and professional knowledge, but at the same time the cooperation of the branches is one of the main sources of innovative scientific and professional solutions. The research activities indirectly contribute to the body of professional knowledge and provide solutions for industrial problems. They are related to different agencies inside and outside the university in terms of resources and proposals. In more detail the research activities are illustrated in Fig. 2.

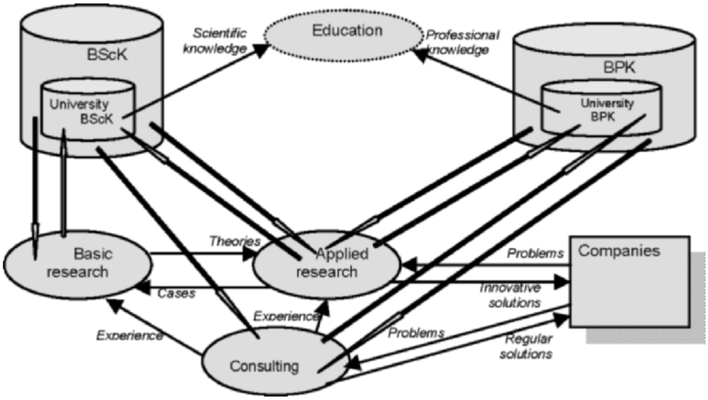


Fig. 2. Knowledge flows (simplified, shown by arrows) in the university research context

At the technical university basic (or pure) research and applied research as well as consultancy influence the bodies of university's scientific and professional knowledge and thus indirectly also the educational process of the university. Quite dense relationships between knowledge bodies, research processes and their external agencies (inside and outside of the university) suggest a necessity for knowledge oriented IS support for research work and its administration at the technical university (further in the text – university).

3 Related work

Necessity to support research work by advanced information technologies is well understood (Lincke 1998, Yao 2003, Ludascher 2005). One of the broadest frameworks for research support systems is given by Yao (2003), who groups information technology tools around seven phases of scientific research, namely: Idea generation, Problem definition, Procedure design/planning, Observation/experimentation, Data-analysis, Results interpretation, and Communication phases. The workflow system that consists of exploring support, retrieval support, reading support, analysing support, and writing support is suggested. The system should provide profile management, resource management and data/knowledge management facilities. It should consist of discipline independent component that supports general research process and domain specific components that support research in specific disciplines. Expert systems, machine learning, data mining, text mining, computer graphics, data visualisation and intelligent information agents are advocated as feasible tools for workflow system's implementation. The framework given in Yao (2003) is quite broad, but it does not show how exactly each phase of research process is supported and how supporting components may be organised in an integrated research support system. The main emphasis in the system is on information retrieval in the early phases of research, data analysis and writing.

Actually, most of research work support systems are devoted to the search of literature, scientific data processing, and scientific knowledge/data sharing. An interesting solution for creation, integration, reviewing and dissemination of domain specific knowledge is aimed by "NetAcademy" approach (Lincke et al. 1998), which uses system-immanent vocabulary allowing the mediation of knowledge through powerful search mechanisms on a semantic level. Information on tools for collaborative data processing and analysis is quite abundantly available (e.g., Watson 2001, Ludascher 2005). Universities can utilise such tools on personal, group, intra-university and inter-organisational levels.

Another branch of related work concerns knowledge management issues of research. The most common issue here, especially for basic research, is ontology oriented knowledge management that helps to handle scientific data (Zhao 2005). For applied research an important tool is collaborative multi-sector knowledge creation that requires a particular knowledge management culture (Rod 2005); and the phenomenon of university spin-offs (Shane 2004, Freeman 2004).

Knowledge management integrated with quality management is described in Rodriguez-Ortiz (2003). The paper states that "in an environment of competitiveness, productivity and quality, the investigation activity is no longer an area where researchers have a wide margin of performance to exploit their creativity, now the delivery schedules and the cost control are factors that have to be considered for success of a research project and the consequent customer of client satisfaction". Researcher's brain is considered as a non-traditional machine that produces the research result. Research project life cycle consists of four phases: Planning and definition, Execution, Delivery, and Evaluation. Knowledge management support for research projects includes the classification and storage of different project related documents such as designs, prototypes, studies, diagnosis, courses, methodologies, as well as proposals, contracts, quality plans, etc. This perspective of knowledge management of research will be discussed in more detail in Section 4.

The financial and non-economic metrics for evaluation of research are discussed in Geisler (2002).

An interesting issue that emerges in modelling of knowledge intensive processes is the possibility to separate knowledge management tasks and normal tasks (Papavassilio 2003). This is a challenge in university research activities context as in this case almost all tasks can be attributed to knowledge management. The analysis of this issue is outside the scope of the paper.

In general one can see that there are issues of research activities support that are well investigated and solution rich, e.g., scientific data processing, workflow support for data analysis, etc. However, in university context, research activities should be viewed taking into consideration knowledge flows between different research types and consultation as well as the impact of research activities on education (see Fig. 1 and Fig. 2 in Section 2). None of above mentioned papers handles these issues in an integrated manner. To move towards the system that can handle all knowledge flows relevant in the university context, in Section 4 we have employed the process perspective of research activities that enables us to deal with all issues of interest reflected in Fig. 1 and Fig. 2.

4 The process perspective of research activities

In this section we start with the process model of research activities. This model is used for further analysis of research activities in the light of APQC and KVC frameworks. At the end of the section we present a simplified conceptual scheme that can serve as a guide for the development of knowledge management oriented IS for research activities support.

4.1 The process model of research activities

The process model of research activities is based on seven-phase research process description of Yao (2003). This framework has been chosen because it itself is the result of analysis of several research activities frameworks. The basic processes and their inputs and outputs are shown in Fig. 3.

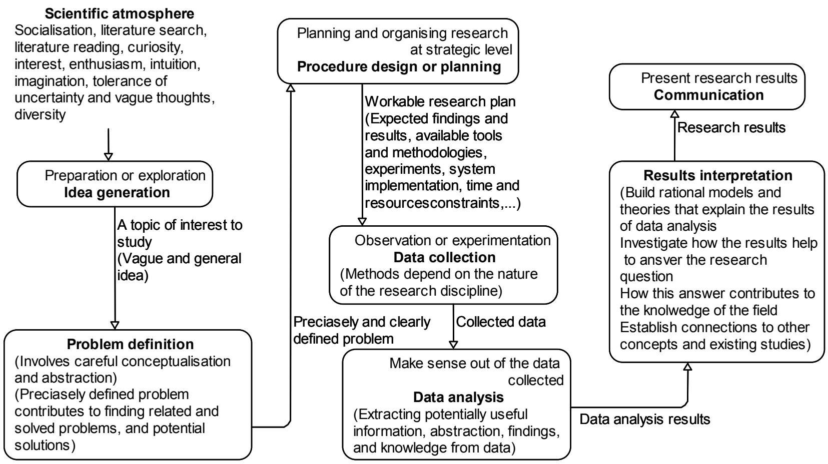


Fig. 3. Business process model of the university research activities

The model includes seven processes, namely Idea generation, Problem definition, Procedure design and planning, Data collection, Data-analysis, Results interpretation, and Communication. Ideally each of those processes as well as the important source of research – scientific atmosphere should be supported by knowledge oriented IS. It is necessary to emphasize that the “research atmosphere” is not always the ultimate trigger of research process (Rodriguez-Ortiz

2003). This refers to research projects invoked by problems proposed by companies, which need help of researchers as well as different calls for research proposals from other agencies (see Fig. 1). In those cases scientific idea and problem definition phases may be influenced by project intended project scope, schedule and metrics (Geisler 2002, Sergeant 2006). Therefore from the point of view of IS support it is reasonable to distinguish between Discovery oriented research and Project proposal oriented research.

4.2 Analysis of research using APQC framework

APQC framework (APQC 2005) is a benchmarking tool for process improvement provided by American Productivity and Quality Centre. The framework includes 12 processes described in high level of detail. They are divided in five operating processes and seven management and support processes. In Table 1 research processes are compared to APQC operating processes on a highest possible level of abstraction.

Table 1. Research processes and APQC operating processes

Research processes	APQC operating processes
Idea generation	Develop vision and strategy
Problem definition	Develop vision and strategy
Procedure design and planning	Design and develop products and services
Data collection	Design and develop products and services
Data analysis	Design and develop products and services
Results interpretation	Design and develop products and services
Communication	Market and sell products and services
	Deliver products and services
	Manage customer service

On the highest level of abstraction initial processes of research are more detailed than the APQC processes, because they show the specifics of scientific activities. On the other hand, research results communication is described by three APQC processes. Viewing research from the production perspective suggests also a high contribution to BPK, which is hardly given by basic research. Thus, IS supported consultation activities (Fig. 2) could compensate the peculiarities of basic research and bring balance between university BScK and BPK.

Correspondence between research processes and APQC operating processes suggest, which industry approved IS solutions could be chosen for each research process.

Management and support processes of APQC framework are as follows:

- Develop and manage human capital
- Manage information technology
- Manage financial resources
- Acquire, construct and manage property

- Manage environmental health and safety
- Manage external relationships
- Manage knowledge, improvement and change

Those processes and their IS solutions are relevant in research support on several levels: the level of individual projects, departmental level, university level, and inter-organisational level in partnership relations.

4.3 Analysis of research using KVC framework

KVC framework (Holsapple and Jones 2004, Holsapple and Jones 2005) presents five primary activities:

- Knowledge acquisition – acquiring knowledge from external sources and making it suitable for subsequent use
- Knowledge selection – selecting knowledge from internal sources and making it suitable for subsequent use
- Knowledge generation – producing knowledge either by discovery or derivation from existing knowledge
- Knowledge assimilation – altering the state of organisation's knowledge resources by distributing and storing acquired, selected or generated knowledge
- Knowledge emission - embedding knowledge into organisational outputs for release into the environment

In Table 2 research processes are compared to KVC operating processes on a highest possible level of abstraction.

Table 2. Research processes and KVC primary processes

Research processes	KVC primary processes
Idea generation	Acquisition, Selection, Generation, Assimilation
Problem definition	Acquisition, Selection, Generation, Assimilation
Procedure design and planning	Acquisition, Selection, Generation, Assimilation
Data collection	Acquisition, Selection, Generation, Assimilation
Data analysis	Acquisition, Selection, Generation, Assimilation
Results interpretation	Acquisition, Selection, Generation, Assimilation
Communication	Emission

The KVC framework does not impose any sequence of the primary activities. One can see in Table 2 that the research processes also do not impose the sequence for those activities (except of Emission, which corresponds to Communication). This suggests service orientation as the appropriate architecture for primary knowledge activities.

Table 2 proves high knowledge intensity of research processes and suggests wide spectrum of knowledge management oriented IS solutions for support of each research process.

The KVC framework presents also four secondary activities: namely, Knowledge measurement, Knowledge control, Knowledge coordination, Knowledge leadership. Those activities partly overlap with the management and support processes of APQC framework and in some extent may be used as organizers of primary activities services (Holsapple and Jones 2005). As KVC framework has been developed by analysing 200 variations of primary activities and 300 variations of secondary activities and is presented in a detailed way, it may serve as a tool for selection of particular knowledge activities for particular university needs.

4.4 Conceptual scheme of relationships between different knowledge bodies

Results of analysis of research from process perspective integrated with the findings of related research as well as features of the context of university research activities suggest the conceptual scheme reflected in Figure 4.

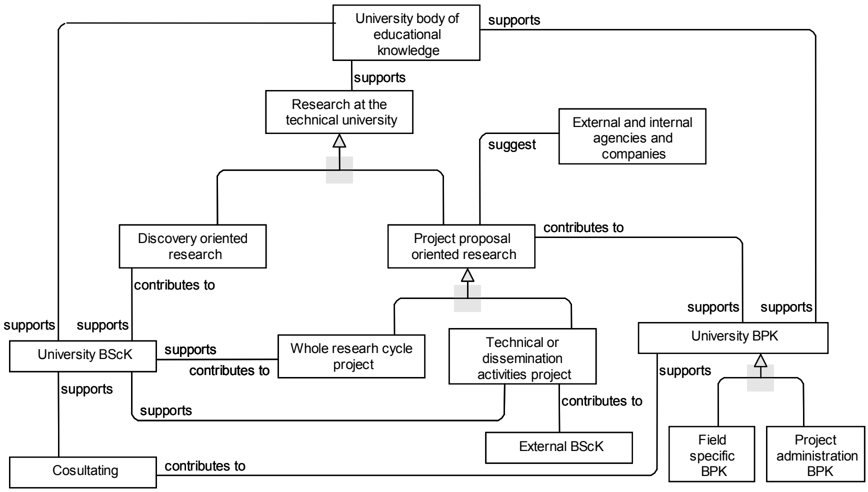


Fig. 4. Simplified conceptual scheme of relationships between different knowledge bodies relevant for research activities

A simplified conceptual scheme in Fig. 4 suggests three different types of research processes: discovery oriented research processes, project proposal oriented processes that generate new scientific knowledge and project proposal oriented processes that do not generate new scientific knowledge (usually those are projects where a university team serves as a technical developer of a particular object or as a disseminator of research results of other universities). The scheme includes also consulting as a supplementary type of research processes that uses scientific knowledge but contributes only to professional knowledge of the university. The scheme also suggests four interrelated bodies of university knowledge, namely BScK, Field specific BPK, Project management specific BPK and Body of educational knowledge. All these bodies of knowledge should be developed and maintained in terms of explicit codified knowledge and supported in terms of tacit knowledge. Thus the information system should support all three types of research, consulting, four types of knowledge repositories, as well as relationships between all entities mentioned and collaboration between researchers at different administrative levels.

5 IS development guidelines for support of research activities at the technical university

In this section we present several guidelines for IS development that are derived from research results presented in previous sections. After each guideline information about the issues, which have led to it, is given in brackets. The guidelines are as follows:

- IS development should be based on well understood research processes that are checked against necessities of knowledge production and knowledge value chain (Sections 4.2 and 4.3, related work (Geisler 2002, Rodriguez-Ortiz 2003)) in terms of supporting IS solutions
- IS should support knowledge management oriented culture (related research (Rod 2005))
- IS should support research atmosphere (Section 4.1)
- IS should support Discovery based and Project proposal based research, and Consultation (Sections 4.1 and 4.4)
- IS should consist of discipline independent component that supports general research process and domain specific components that support research in specific disciplines (related work (Yao 2003))
- IS should support four interrelated knowledge repositories (scientific, field specific professional, administrative professional, and educational) (Section 4.4) and tacit knowledge flows that support these repositories.
- IS should provide personalization support (related work (Yao 2003))
- IS should provide collaboration support (Section 3)
- IS should support metrics based evaluation procedures (KVC, APQC frameworks and related work (Geisler 2002, Rodriguez-Ortiz 2003))
- Primary activities of KVC framework and their support by IS components may be organized in service oriented architecture (because there are many to many relationships between research processes and primary activities (Section 4.3) as well as between primary activities and supporting IS components (Yao 2003))
- Hypothetically knowledge production activities and corresponding IS solutions should be organised in fractal architecture (as they have similar patterns at different levels of abstraction (Section 4)).

6 Conclusions

There are many solutions of IS support for research activities. Most efforts in this area have been directed to scientific data management and collaboration. However, in university environment, additional issues are to be considered when introducing IS for support of research activities, namely: the impact of research on education and industry oriented dimension of research activities. Consideration of these issues suggest knowledge management and process oriented approach of IS development where IS primarily supports all processes of knowledge production and invokes all knowledge activities that add value to scientific knowledge produced by research.

The guidelines given in this paper are the result of theoretical research and are intended to be tested in the development of the conception of IS support for research activities at the technical university.

Acknowledgment

Work on this paper is partly supported by research grant No. U7121 of Latvian Ministry of Education and Science and Riga Technical University.

References

APQC (2005) Process Classification Framework, APQC. Available at Http://www.apqc.org, accessed April 28, 2006.

Fadel KL and Tanniru M (2005) A knowledge centric framework for process redesign. In: SIGMIS-CPR'05, April 14-16, 2005, Atlanta Georgia, ACM 1-59593-011-6/05/0004, pp 49-58.

Freeman R, Tang K, Vohora A (2004) Taking Research to Market: How to Build and Invest in Successful University Spinouts, Euromoney Institutional Investor.

Geisler E (2002) What do we know about: R&D metrics in technology-driven organisations. Http://cims.ncsu.edu/documents/rdmetrics.pdf, accessed April 27, 2006.

Holsapple CW and Jones K (2005) Exploring secondary activities of the knowledge chain. Knowledge and Process Management, Vol. 12, No. 1, pp 3-31.

Holsapple CW and Jones K (2004) Exploring primary activities of the knowledge chain. Knowledge and Process Management, Vol. 11, No. 3, pp 155-174.

Hornbostel St (2006) From CRIS to CRIS: Integration and interoperability. Available at the website of Current Research Information Systems conference (CRIS 2006): Http://ct.eurocris.org/CRIS2006/, accessed June 28, 2006.

Lincke D-M, Shmidt B, Schubert P, Selz D (1998), The NetAcademy: A novel approach to domain-specific scientific knowledge accumulation dissemination and review, 1060-3425, IEEE

Ludacher B, Altintas I, Berkley Ch, Hoggins D, Jaeger E, Jones M, Lee EA, Tao J, Zhao Y (2005) Scientific workflow management and the Kepler system. Concurency and Computation: Practice and Experience, 2005, Wiley Interscience DOI: 10.1002/cpe.994

Papavassiliou G, Ntiodis Sp, Abecker A, Mentzas Gr (2003) Supporting knowledge-intensive work in public administration processes. Knowledge and process Management, Volume 10, Number 3, pp 164-174, Wiley, 2003.

Rod MRM (2005) Collaborative multisector knowledge creation – the Institute of Health Economics. Available at Http://kmap2005,vuw.ac.nz/papers/, accessed April 27, 2006.

Rodriguez-Ortiz G (2003) Knowledge management and quality certification in research and development environment. In: Proceeding of the Fourth Mexican Conference on Computer Science (ENC'03), 0-7695-1915-6/03 IEEE.

Sergeant DM (2006) Using a virtual research environment to present CRIS grouped to support the real research users' research lifecycle. Available at the website of Current Research Information Systems conference (CRIS 2006): Http://ct.eurocris.org/CRIS2006/, accessed June 28, 2006.

Shane ScA (2004) Academic Entrepreneurship: University Spinoffs and Wealth Creation, Edward Elgar publishing.

Verhoef R and Qureshi S (2005) Collaborative infrastructures for mobilizing intellectual resources: Assessing intellectual bandwidth in a knowledge intensive organisation. In: Proceedings of the 38th Havaii International Conference on Systems Sciences, 0-7695-2268-8/05 2005 IEEE.

Yao YY (2003) A framework for Web-based research support systems. In: Proceedings of the 27th Annual International Computer Software and Application Conference (COMPSAC'03), 0730-3157/03, 2003, IEEE.

Zhao Zh, Belloum A, Sloot P, Hertzberger B (2005) Agent technology and scientific workflow management in an e-science environment. In Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05), 1082-3409/05, IEEE.

An Architecture for Highly Available and Dynamically Upgradeable Web Services

Nearchos Paspallis and George A. Papadopoulos

Department of Computer Science, University of Cyprus 75 Kallipoleos Street, P.O. Box 20537, CY-1678, Nicosia, Cyprus
{nearchos, george}@cs.ucy.ac.cy

Abstract: Developing distributed application architectures characterized by high availability has always been a challenging and important task both for the academic and the industrial communities. Additionally, the related requirement for dynamic upgradeability is usually examined within the same context as it also aims for high availability. Although a number of architectures and techniques have been proposed and developed for improving the availability and upgradeability of traditional distributed systems, not many of them are directly applicable to Web service-based architectures. Recently, Web services have become the most popular paradigm for business-to-business and enterprise application integration architectures, which makes their availability increasingly important. This paper builds on existing high availability and dynamic upgradeability techniques which can be applied to Web service-based systems. Based on them it describes an architecture which enables high availability and dynamic upgradeability both for newly developed and for prefabricated Web services.
Keywords: Dependable systems, Service engineering.

1 Introduction

Web services are the technology-of-choice for interoperability within non-homogeneous systems. They are briefly defined as *"self-contained, modular applications that have open, Internet-oriented, standards-based interfaces"* [3]. Although many expect that Web services will change the way enterprises interoperate with each other in the long-term, they have already proven themselves very useful in solving many of the interoperability problems that have troubled application integration efforts.

Today, one can observe more and more enterprises depending on their Web accessible services to a continuously increasing degree. Some of these services are critical and the enterprises invest a lot of effort and resources in maintaining them as highly available as possible. Examples of Web services requiring high levels of availability include medical, stock market, and airline ticket reservation applications.

While a number of techniques target at maintaining the high availability of a service in the exceptional case of a fault, additional effort is also required to maintain the accessibility of the service when it is undergoing a scheduled update. This is why systems designed for high availability usually provide mechanisms enabling dynamic (also known as live) software upgrades as well. Typical reasons for upgrading software include bug fixes, functionality enrichment and performance enhancements. Naturally, highly available services are expected to be upgradeable in a safe and consistent manner.

This paper studies techniques for achieving high availability of services offered over the Web and also techniques that allow dynamic upgrades of those services. Although some of these approaches could be applied to general distributed systems, this work concentrates on Web services and their underlying technologies (e.g. SOAP, WSDL, UDDI, etc).

In the next section, we examine the Web services technology from the availability and upgradeability point of view. Then, existing techniques for high availability and dynamic upgradeability are presented in section 3. Following that, a methodology for improving on the availability and upgradeability of Web services is proposed in section 4. The proposed method involves automatic generation of stubs and skeletons using the WSDL document. Furthermore the same section argues on the advantages and the disadvantages of this approach and, finally, section 5 summarizes the conclusions and points to future work.

2 Web services

Web services can be described as applications accessible over the Web [3]. More accurately the World Wide Web Consortium (W3C) describes them as: "*…software applications identified by URIs, whose interfaces and bindings are capable of being defined, described and discovered as XML artifacts. Web services support direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols*."

The promise of Web services is to serve as the foundation for a new generation of business-to-business (B2B) and enterprise application integration (EAI) architectures. Because Web services are both language and platform neutral, it is a common practice for enterprises to expose a selected subset of their functionality (of newly developed but also of legacy information systems) as Web services. This ability greatly contributes to the popularity of Web services in both B2B and EAI scenarios.

2.1 Highly available web services

In this paper the availability refers to a measure of the fault tolerance of a Web service. Consequently, high availability is defined as a goal that we try to achieve by employing a number of methods and techniques. Literally, availability is the percentage of time a Web service is available and functioning within its operational requirements. Obviously, to achieve high availability a service needs to maximize its uptime and minimize its downtime.

High availability is important for many enterprises because their system responsiveness is directly related to their customer satisfaction and, consequently, to their operations turnover. Also for many enterprises which are highly depended on Web services, their downtime is usually proportional to significant revenue losses. In many other applications such as medical information systems, high availability is inherently critical and extremely important.

The availability can be affected by factors such as connectivity (i.e. network availability) and server failures (i.e. hardware and software faults). Thus any solution aiming to provide reasonable protection against failures should cope with both factors.

Based on their definition, Web services need to be discoverable and also facilitate interactions with other systems by continuously allowing binding and interaction. Therefore, to ensure that Web services maintain high availability, their discovery and binding functionality need to be enhanced with appropriate mechanisms. The discovery of Web services is generally performed using service directories based on the UDDI standard, which are Web services themselves. Thus, improving on the availability of general services offered over the web, consequently benefits the discovery of Web services as well.

2.2 Dynamically upgradeable web services

With dynamic upgrades, we refer to the replacement of software components at runtime, with minimal (preferably zero) service interruption.

In the client-server paradigm, the upgrade can take place at either of the two sides, or at both. In most cases, client-side upgrades are more straightforward compared to server-side upgrades because they can take place in a controlled manner (i.e. the client can be instructed to suspend or drop any connections to the server or even completely shut an application down if necessary.) This eases the task of replacing some components or the whole application.

Contrary to this, upgrading server side components is significantly more challenging. Because no pre-determined downtime is known, clients initiate transactions with the service in an arbitrary way. In [9] Kramer *et al.* introduced the notion of quiescence, i.e. a period within which the component can be safely upgraded (e.g. replaced.) A component is said to be quiescent when the component itself and all components linked to it are in a passive state, i.e. seize initiating but continue serving transactions.

Dynamic upgrades are required for a number of reasons. These are classified as corrective, perfective and adaptive [11]. Corrective upgrades are used for fixing bugs (e.g. discovered after deploying the service). Perfective upgrades are used to enhance the product functionality and performance and, finally, adaptive changes are needed for adjusting services to a changing environment.

Dynamic upgradeability is important because it enables continuous service operation in the event of scheduled upgrades. Thus mechanisms for dynamic upgradeability are important (and quite often required) supplements to high availability architectures.

3 Related work

This section reviews existing high availability and dynamic upgradeability techniques with emphasis on those targeting server-side faults. While complex upgrade mechanisms have been proposed, a common approach employed by high availability architectures includes temporary redirection of the traffic before upgrading the server, and then redirection of the traffic back to the original server once that is completed.

It is worthwhile mentioning that different techniques operate at different layers of the system architecture. At the lowest layer some techniques use hardware replication, while at the highest layer other techniques simply embed the mechanisms required for high availability into the applications themselves.

There are a number of criteria that can be considered while evaluating architectures. Here, we concentrate on the transparency of the investigated techniques and their applicability to prefabricated Web services. By transparent we refer to those techniques which can be applied without requiring any major changes to existing infrastructures, i.e. those that can be directly applied to existing Web services.

3.1 Existing techniques

In [4] Birman *et al.* discuss methods for adding high availability to Web services. In addition, they also discuss how to enable autonomic behavior i.e. how to enable servers to automatically discover and configure themselves and then operate securely and reliably in an automated manner.

Their work uses extensions to the Web services model which aim to support standard services for monitoring the health of the system, self-diagnosis of faults, self-repair of applications and event reporting. The solution builds on existing technologies, such as WS-Transactions [6] and WS-Reliability [8] but it eliminates their need to save data to persistent memory or wait for failed components to restart.

This solution can act as a router component of a Web service platform, making it suitable for providing transparent high availability to existing applications.

However this approach does not define explicit methods to enable dynamic upgrades.

In [7] Cotroneo *et al.* propose an architecture which improves the availability of web-based services, such as Web servers, FTP servers, and video-on-demand servers. Their work examines the problem from a Quality of Service (QoS) perspective and specifically targets real-time systems. Their architecture provides application developers with an alternative API which can be used to access the network instead of the typical communication libraries.

Clearly, this approach is not transparent to the developers as the provided API is used instead of the standard UNIX network socket libraries. This solution operates at the network and the operating system layers and thus cannot be applied to other platforms.

In [12] Vilas *et al.* present a work where high availability is achieved at the Web service layer. Their proposed technique introduces the notion of Virtualization. This technique creates new virtual Web services and exposes them to the clients instead of the actual ones. At the back-end, the real Web services are invoked while they are internally managed in a cluster.

The authors of this work define three requirements: detecting faulty servers, providing maintenance mechanisms for the cluster and providing mechanisms for adding and removing servers in the cluster as needed.

Virtualization is a common technique with existing and popular applications in related fields such as in web servers. The way it works in the case of Web services is by grouping one or more services inside a unique wrapper which is then published as a single, standard Web service. The clients then use this virtual Web service as if they were contacting the real one.

This approach requires that the developer defines a Virtual Web Service (VWS,) and a VWSDL document. Also a VWS engine is required to enable clustering and high availability. Depending on the complexity of the application, the VWS engine can be as simple as some specialized code in the stub or as complex as a dedicated server. Furthermore, additional techniques are needed for forming and managing the cluster.

In addition to the other works presented in this section, this work does also not provide explicit mechanisms facilitating dynamic software upgrades. In [1] Ajmani provides a thorough and comprehensive list of software upgrade techniques for distributed systems. He starts his review from Bloom’s work on reconfiguration in Argus [5] and continues with recent technologies used in modern systems (such as the Red Hat OS) and also by popular services (such as in Google’s infrastructure).

3.2 Evaluating existing techniques

Although all the techniques presented in this section can directly or indirectly, improve the availability of Web services, no two of them are equal with respect to their development requirements. In principle, we are interested in techniques that can be applied in general situations without any specific requirements regarding the *programming languages*, *infrastructures*, or *hardware*.

Ideally, a solution would allow automatic deployment and management of Web services and transparently improve on their availability. Apparently, such a solution should be applicable to prefabricated Web services. Furthermore, it should allow dynamic upgrades of the deployed software, preferably with minimal, if not zero, interference to the service. To the best of our knowledge, none of the presented or existing techniques fully satisfies all these requirements. In the next section we propose an architecture that can provide the basis for delivering a solution which meets all these criteria.

4 An architecture for high availability and dynamic upgradeability

This section studies the requirements for a system offering both *high availability* and *dynamic upgradeability*. Then, it proposes an architecture which is designed to meet these requirements and transparently improve on both the availability and upgradeability characteristics of prefabricated Web services. Finally, this section concludes with a discussion on the drawbacks and the benefits of the proposed design.

4.1 Requirements for high availability and dynamic upgradeability

First, the high level requirements for architectures targeting high availability are detected and enumerated. These requirements are then further complemented with additional ones targeting dynamic upgradeability, as the latter is argued to be a key requirement for improving availability.

Mathematically, availability is simply defined as the ratio of time during which the service is considered to be satisfying to the service consumer. Of course, defining when a service is satisfying is not trivial and it requires further clarification. For example, in some cases a service response of a few minutes might be acceptable, while in others sub-second responses are essential.

Detect when the service responsiveness becomes unsatisfactory. The first requirement is to detect when the service responsiveness becomes unsatisfactory to the clients. A detection mechanism must be used to detect these events and inform the appropriate components. Once deviation outside the accepted operation range is detected, a procedure is initiated which aims to resume the service. Consequently, the second requirement is to carry out the necessary actions required to restore the service normal operation within the predefined boundaries.

Manage the availability infrastructure. The second requirement is the ability to manage the availability infrastructure. For example specific architectures might need to define the order of the servers in the failover list, modify the set of servers in the cluster, or change the monitoring attributes and characteristics. In this paper we focus on the first two requirements. More management requirements are expected to be considered in future work.

The high availability technologies can be classified into those that failover on the server side, and those that failover on the client side. In the first case the classic cluster-based solution is the obvious approach, herein referred to as intra-enterprise availability. In this case a cluster of servers appears as a single server, continuously offering the service at a predefined IP address, even in the event of single server failures. This is the common case, where the clients are completely unaware of any failures or possible actions that were taken to recover the system back to fully operational mode. Naturally, in this case the clients will not be able to recover from any network outages, regardless of the cluster health.

In the second case, the client is designed to be more adaptive with regards to availability. More specifically, if a service failure is detected (and not recovered within some predefined time) the client initiates a failover procedure to another service, possibly provided by a different enterprise. We refer to this technique as inter-enterprise availability.

The mechanism for discovering and selecting a Web service in this case can be similar to that of a typical UDDI registry. More than one UDDI registries can be contacted for better fault tolerance and for a richer options pool. Additionally, this method requires specialized mechanisms embedded in the client stub and additional logic might also be necessary to ensure that the failover involves a semantically and functionally equivalent Web service. The latter is a challenging issue because additional meta-information with regards to the service provider (e.g. pricing) might be needed when deciding on a suitable alternative service to failover to.

In the first case where the service usually runs on top of a server cluster, it is necessary to use a mechanism that continuously monitors the health of the individual servers of the cluster. In this way, any possible failures are detected before they get noticed by the clients. The failover can be performed using any of the existing methods proposed so far.

In the case of inter-enterprise availability, the detection and failover mechanisms must be embedded into the client-side. This method adds significant complexity into the clients, but has the advantage of surviving long-running network outages that prevent communication with the server side.

Support for dynamic upgrades. The last requirement we consider is the support for dynamic upgrades. By dynamic we refer to upgrades that take place at *runtime*, preferably without any service interruption. The upgrades can take place at any of the client, the server, or both sides. The following paragraphs examine the dynamic upgrade-related requirements in detail, building on results described in [2].

First a management mechanism that instructs the nodes when to upgrade must be defined. Consider for example the case where the upgrade of a service running on a cluster (i.e. for increased availability) is required. Apparently, not all the servers can be upgraded simultaneously because that would compromise the service's availability. A management mechanism can control how the servers are upgraded, so that a set of servers is consistently operational with an acceptable level of availability.

The second requirement is to provide a way to control when the servers are upgraded. Although the most straightforward solution would be to arbitrarily remove the node from the cluster and upgrade it (letting the availability infrastructure take care of the interrupted transactions) it is not an optimal one. A more appropriate solution would be one detecting an appropriate time-frame within which the upgrade would be possible without any service interruption and without breaking the consistency of the system.

The third requirement is to provide mechanisms that guarantee the normal operation of the system when nodes are running different versions of software. If, for example, the server is upgraded to support a different set of operations (e.g. specified by a different WSDL document), appropriate adaptation of the invocations is needed until the clients are also upgraded to the latest version.

The last requirement mandates a way to preserve the persistent state of servers from one version to another. If, for example, the client is in the middle of executing a long process consisting of multiple operations, it is important that the upgraded software preserves its state and continues with the next operation in the process after the upgrade is completed (rather than having to restart a large computation task). This applies to both the client and the server sides.

4.2. Smart-stubs and smart-skeletons

In order to satisfy all the requirements we have specified, we propose a skeleton architecture where components can be added and existing techniques be reused. This architecture supports prefabricated Web services and it builds on a minimal model described in [3] and depicted by Figure 1.

In this architecture the WSDL document is used as input to specialized compilers which generate client-side and server-side proxies, typically referred to as *stubs* and *skeletons*. A different compiler is required for each of the client and the server side. The application objects can then bind to the proxies and invoke the operation defined in the WSDL documents. These proxies enable distributed communication with the use of SOAP-based messages.

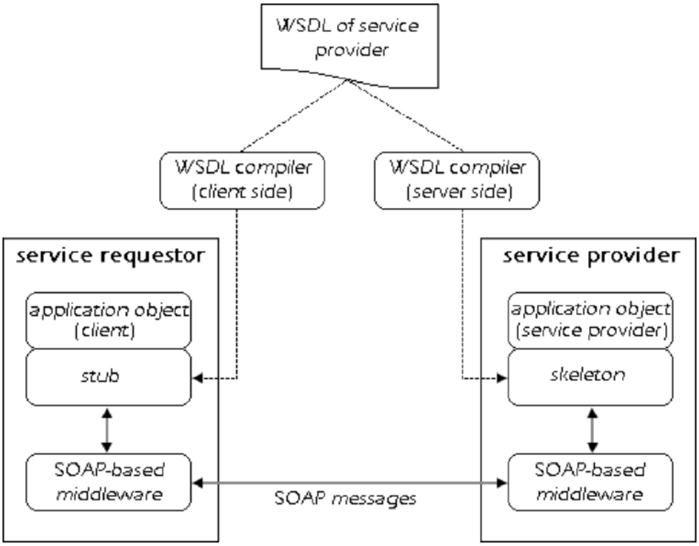


Fig. 1. Typical Web Service interaction: Based on the given WSDL document, appropriate proxy objects (i.e. the stub and the skeleton) are generated which are then used to facilitate the communication between the distributed objects by abstracting the remote objects as local

This approach extends the idea expressed in [10], where the authors argue that the reliance on machine readable metadata is probably one of the key defining aspects of Service Oriented Architectures (SOA). In this approach, the middleware exploits the additional metadata (in the form of availability directives and preferences) to enable seamless enhancements to the overall service availability.

The presented architecture requires the dynamic generation of intelligent proxies, namely smart-stubs and smart-skeletons. These proxy components directly accept invocations from the application objects in a fashion similar to the standard Web service paradigm. The proposed architecture is depicted by Figure 2. The grayed-out areas illustrate deviations from the original architecture.

In Figure 2, the HA-related properties are used to provide information describing different aspects of the high availability-related functionality such as connection time-out, preferred failover list, etc. These properties are then encoded into the generated smart-proxies.

Special compilers (i.e. HA-aware WSDL compilers) are used for the generation of smart-stubs and smart-skeletons. In addition to providing the functionality for SOAP–based communication these proxies contain additional functionality for dealing with rerouting, blocking and adapting SOAP messages.

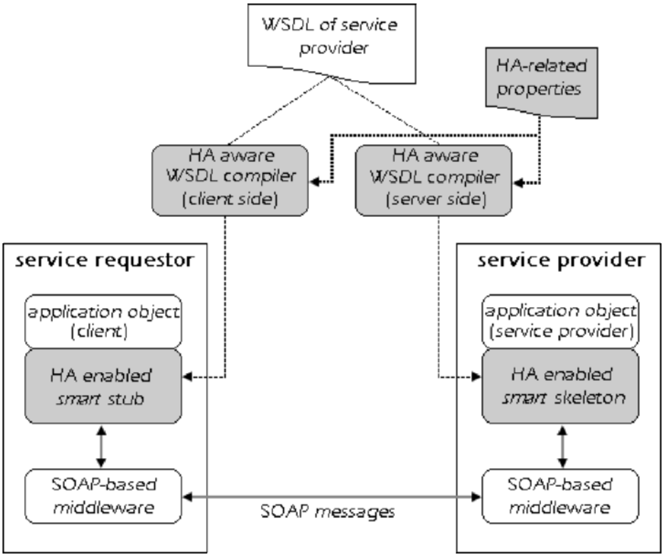


Fig. 2. Web Service interaction using smart-stubs and smart-skeletons: Based on a given set of properties describing the high availability requirements (or strategy), specialized WSDL compilers generate the smart-proxies (i.e. the smart-stub and the smart-skeleton). These objects act in a manner similar to that of the normal proxies, but additionally they incorporate specialized code which allows them to improve on the availability of the service, i.e. by enabling automatic failover and load-balancing

The smart-stubs and the smart-skeletons are thin, automatically generated proxy components. Their role is to implement the logic required to allow automatic and seamless rerouting of SOAP messages in order to ensure high availability in the event of faults. Additionally, their role includes blocking and adapting SOAP messages in order to enable seamless dynamic upgrades of software.

To make the processing and handling of the SOAP messages transparent to the end-users, these proxies intercept the communication on both the client and the server sides and appropriately reroute, block and adapt the communicated messages.

In addition to encoding the invocations to SOAP messages and marshalling or un-marshalling the data arguments, the smart-stubs provide additional logic for handling failures on the server side (e.g. by failing over to another server). Similarly, the smart-skeletons provide functionality for blocking messages while upgrading a Web service and also for adapting messages targeting a different version of the deployed object.

4.3 Satisfaction of the requirements by the architecture

The following paragraphs discuss how the general architecture, described here, satisfies the requirements that were detected in the previous section. First, in order to be able to discover when the service responsiveness becomes unsatisfactory, special client-side code is embedded in the smart-stub. This code allows the detection of faults and enables the failover to different service providers. Typically, this code is based on existing techniques which have a proven and trusted track in the area of fault-detection. In their simplest form these techniques usually depend on preset response-deadlines or on health monitoring systems which actively and periodically contact the servers (i.e. poke) to detect if they are responsive (i.e. healthy).

To manage the availability infrastructure the smart-proxies embed specialized functionality. Their management can be based either on static, predefined strategies encoded in the input data (i.e. in the HA-related properties), or it can based on a more dynamic and interactive scheme. The latter implies that the smart-proxies, which can serve as interception points, could be exploited to block, reroute and generally manage the operation of the Web-service from an availability point-of-view.

Additional logic might also be required to ensure the correctness of protocols such as the WS-Transaction. In particular, if the client decides to failover to another service provider while processing a business activity, specialized actions are required to ensure that suitable compensation operations are issued on the original service provider when it returns back online.

Last, dynamic upgradeability requires support by both the smart-stub and the smart-skeleton components. In particular they should both include code to address the additional, refined requirements that have been detected for enabling dynamic upgradeability.

For the management mechanism either an external, centralized coordination server should be used or special code should be embedded into the smart-stubs (or equivalently into the smart-skeletons.) Clearly, the first approach is more straight-forward from an implementation point-of-view. Embedding the code in the smart-proxies has the apparent advantage of making the system more self-reliant (but also more complicated). Finally, in simple scenarios where only a single client (or server) is upgraded, the management mechanism could be unnecessary.

For detecting when it is appropriate to perform an upgrade, specialized code should again be embedded in the implementation of the smart-stubs (or smart-skeletons). This requirement is usually related to the persistent state requirement. Suitable solutions exist which simultaneously address both. The latter usually requires that the upgraded applications provide mechanisms for enabling state persistence across different versions as well.

Finally, the concurrent support of different versions is addressed. If both the new and old versions of Web services define the same operations (i.e. they are described by the same WSDL document) then there is no need for any adaptation. If the two versions define different operations though, adaptors are required in the smart-proxies to map the old-version invocations to the corresponding operations of the new version. This is something that can be directly reused from existing solutions (i.e. designed to enable dynamic upgrades) and embedded into the smart-proxies.

Of course, this architecture is a high level overview of a skeleton system, purposely designed to intercept Web service communication at the point where invocations are applied. In this way the provided mechanisms have maximum control on the invocations and block the communication of SOAP-messages when necessary (e.g. when upgrading the Web service). Still, some issues remain to be addressed before this architecture fulfils the detected requirements.

5 Conclusions and future work

Custom solutions enabling high availability and dynamic upgradeability can be prohibitively complex and costly. In addition, they also require a combination of technologies and services such as *disaster recovery*, *consulting*, *assessment* and *management*. This paper concentrates on technical aspects of high availability and in particular on scenarios where Web service failover is used to maximize their availability. Additionally, it describes reusable techniques aiming at continuous availability during dynamic upgrades of Web services as such techniques are also required by any high availability framework.

The contributions of this paper are twofold. First, the need for availability and upgradeability is identified and existing techniques used to tackle the problem in Web service-based systems are presented. Second, a general architecture is presented which can provide the basis for systems aiming at high availability and dynamic upgradeability.

A main contribution of this architecture is that it improves on the availability of Web services, even in the event of inter-enterprise failover. As the failover mechanism entirely resides within the client this approach allows failing over to a Web service provided by a completely different entity. Also, the proposed architecture can be of benefit to prefabricated Web services as well as it only requires (re)compiling their WSDL documents to generate all that is required: the smart-stubs and the smart-skeletons. Because these smart-proxies are dynamically generated, the proposed architecture is also transparent to both the end-users, and the WSDL developers.

For the future, we plan to more elaborately define the structure and the functionality of the WSDL compilers as well as of the corresponding SOAP messages required by the protocols. Additionally, a prototype implementation is scheduled in order to evaluate the proposed architecture with the development of case study applications. Finally, related cluster management mechanisms will be examined and the use of UDDI directories for advertising and discovering alternative Web service providers will be more thoroughly studied.

6 Acknowledgements

This work was partly funded by the European Union as part of the IST MADAM project (6th Framework Programme, contract number 4169).

References

1. S. Ajmani, "A Review of Software Upgrade Techniques for Distributed Systems", *http://www.pmg.lcs.mit.edu/~ajmani/papers/review.pdf*, 2002.
2. [S. Ajmani, B. Liskov and L. Shrira, "Scheduling and Simulation: How to Upgrade Distributed Systems", *9th Workshop on Hot Topics in Operating Systems (HotOS 2003)*, USENIX 2003, Lihue (Kauai), Hawaii, USA, 2003, pp. 43-48.
3. G. Alonso, F. Casati, H. Kuno and V. Machiraju, "Web Services: Concepts, Architectures and Applications", *Springer-Verlag*, 2004.
4. K. P. Birman, R. V. Renesse and W. Vogels, "Adding High Availability and Autonomic Behavior to Web Services", *26th International Conference on Software Engineering (ICSE 2004)*, IEEE Computer Society 2004, Edinburgh, United Kingdom, 2004, pp. 17-26.
5. T. Bloom and M. Day, "Reconfiguration in Argus", *International Conference on Configurable Distributed Systems (CDS 1992)*, London, England, 1992, pp. 176-187.
6. W. Cox, F. Cabrera, G. Copeland, T. Freund, J. Klein, T. Storey and S. Thatte, "Web Services Transaction (WS-Transaction)", *http://dev2dev.bea.com/pub/a/2004/01/ws-transaction.html*, 2004.
7. D. Cotroneo, M. Gargiulo, S. Russo and G. Ventre, "Improving the Availability of web services", *22nd International Conference on Software Engineering (ICSE 2002)*, Orlando, Florida, USA, 2002, pp. 59-63.
8. C. Evans, D. Chappell, D. Bunting, G. Tharakan, H. Shimamura, J. Durand, J. Mischkinsky, K. Nihei, K. Iwasa, M. Chapman, M. Shimamura, N. Kassem, N. Yamamoto, S. Kunisetty, T. Hashimoto, T. Rutt and Y. Nomura, "Web Services Reliability (WS-Reliability) version 1.0", *http://www.oracle.com/technology/tech/webservices/htdocs/spec/WS-ReliabilityV1.0.pdf*, 2003.
9. J. Kramer and J. Magee, "The Evolving Philosophers Problem: Dynamic Change Management", *IEEE Transactions Software Eng*ineering, 16 (1990), pp. 1293-1306.
10. [N. K. Mukhi, R. Konuru and F. Curbera, "Cooperative Middleware Specialization for Service Oriented Architectures", *13th International World Wide Web Conference (WWW2004)*, New York, NY, USA, 2004, pp. 206-215.

11. [P. Oreizy, N. Medvidovic and R. N. Taylor, “Architecture-Based Runtime Software Evolution”, *20th International Conference on Software Engineering (ICSE 1998)*, IEEE Computer Society, Kyoto, Japan, 1998, pp. 177-186.
12. J. F. Vilas, J. P. Arias and A. F. Vilas, “High Availability with Clusters of Web Services”, *6th Asia-Pacific Web Conference (APWeb 2004)*, Springer-Verlag, Hangzhou, China, 2004, pp. 644-653.

Distributed Service Development in Personal Area Networks

Miklós Aurél Rónai, Kristóf Fodor, Gergely Biczók, Zoltán Turányi, and András Valkó

Ericsson Research, Traffic Lab, 1300 Budapest, 3., P.O. Box 107, Hungary,
Miklos.Ronai@ericsson.com

Abstract: This paper presents the detailed description of the Middleware for Application Interconnection in Personal Area Networks (MAIPAN), which is designed to ease distributed service development for mobile and nomadic environment. This middleware provides a uniform computing environment for distributed applications that operate in dynamically changing personal area networks (PANs). MAIPAN hides the physical scatteredness and device configuration of the PAN and presents its capabilities as a single computer towards the applications. The solution provides easy set-up of PAN-wide applications utilizing multiple devices and allows transparent redirection of ongoing data flows when the configuration of the PAN changes. The proposed middleware interconnects services offered by applications running on different devices by creating virtual channels between the input and output outlets of the applications. Channels can be reconfigured when configuration or user needs change. In contrast to the approaches found in the literature, MAIPAN is a solution where session transfer, dynamic session management are tightly integrated with strong and intuitive access control security. A prototype implementation demonstrates the capabilities of the middleware.

1 Introduction

The ever-growing number of wireless terminals, such as smart phones, personal digital assistants (PDAs) and laptops, raises the need to set up, configure and reconfigure personal area networks (PANs) in an easy and ergonomic way.

This paper describes in details the Middleware for Application Interconnection in Personal Area Networks (MAIPAN) that hides the individual devices participating in the PAN and presents the capabilities of applications running on the devices as if they were located on a single computer. This provides a standard "PAN programming platform", which allows the easy set-up of personal area networks and dynamic connection and disconnection of distributed applications running in the PAN. Application programmers using the uniform application programming interface (API) offered by the middleware can develop software without taking care of the various PAN configurations or PAN dynamics. They can assume certain capabilities, but disregard whether these capabilities are provided by one application running on one device or by a set of applications running on several devices. They only have to register the inputs and outputs of their applications in the middleware, and they do not have to take care of which kind of devices or applications will be connected to these outlets and will use their programs.

The presented middleware contains access control, flexible session management and transferable session control solutions. The middleware contains some intelligent functions, as well, which helps the user to control the PAN and improves human computer interaction (HCI). In theory all kind of solutions for service discovery, physical, link or networking layers can be used with MAIPAN. However, currently MAIPAN is implemented on top of TCP/IP.

The paper is organized as follows. Section 0 is about related work; in Section 0 the basic concepts and the middleware's architecture are outlined. The API is described in 0 and the internal operation of the middleware (e.g., message exchanges, control functions and access control mechanisms) is detailed in Section 0. Finally, Section 0 concludes the paper.

2 Related work

Middleware are essential part of pervasive and mobile computing environments. Mascolo et al. in 1 discussed why traditional middleware (such as CORBA 2) is not well suited for mobile environments and how a mobile computing middleware should be designed. Nowadays several projects are running in these research topics, some of them are presented in the followings.

The goal of the AURA project 3 is to provide each user with an invisible aura of computing and information services that persists regardless of location. The project Gaia 4 designs a middleware infrastructure to enable active spaces in which data and tasks are always accessible and are mapped dynamically to convenient resources present at the current location of the user. The Oxygen project 5 aims to develop very intelligent, user-friendly and easy-to-use mobile devices enabling users to communicate with the system naturally, using speech and gestures that describe their intent. In the frame of the Portolano project the one.world architecture 6 is designed, which is a comprehensive framework for building pervasive applications. The Cortex project 7 addresses the emergence of a new class of applications that operate independently of human control. The key objective of the EasyLiving 8 project is to create an intelligent home and work environment. The Speakeasy approach 9 focuses on the specification of minimal interfaces between devices using mobile agents and mobile code. The Virtual Device 10 concept considers all autonomous devices in the user's personal area network as one big virtual device having multiple input and output units, thus providing a coherent and surrounding interface to the user.

Similar to the Virtual Device concept, MAIPAN represents an entire personal area network as a single device to applications. On the other hand, MAIPAN represents a novel approach in its secure access control mechanism and the use of a transferable control role. MAIPAN access control ensures 1) seamless interworking of various devices of the same user, 2) protection of one user's devices from devices of another user, 3) still enabling controlled communication and lending between devices of different users. MAIPAN manages device access and configuration via a convenient central control entity, the dispatcher. MAIPAN is also unique in enabling the change of the dispatcher role, that is, the session control rights can be transferred between devices, from the old dispatcher to a new one.

The basic ideas of MAIPAN were introduced in 11 and the application programming interface is described in 12. Now, in this paper we provide a detailed description of MAIPAN's internal operation, including its structure, session creation, dynamic session management, session control transfer and access control functions.

3 MAIPAN – Middleware for Application Interconnection in Personal Area Networks

3.1 Basic concepts and definitions

MAIPAN distinguishes among devices, applications and services. The word "device" refers to the physical device and by "application" we refer to the software that offer the "services". For example, using these abstractions in case of a mouse we can say, that the mouse is a "device" where a "mouse application" is running, which offers a "mouse service".

MAIPAN is based on three concepts (**Fig. 1.**): pins, channels and sessions. Applications offering the services have input and output outlets, which are called pins—borrowing the expression from the integrated circuit world. Pins are the connection points of the applications to the middleware, so the middleware sees the applications in the PAN as a set of input and output pins. A pin has a predefined type, which shows the type of data that the pin can emit or absorb, that is, the type of information the application can handle (e.g., mouse movements, keystrokes). According to the needs new types can be defined any time. The dispatcher application is responsible to connect the pins with appropriate types.

To enable communication between pins, the middleware creates and reconfigures channels, which are point-to-point links that interconnect pins. The set of channels that are necessary to use a PAN service is called session.

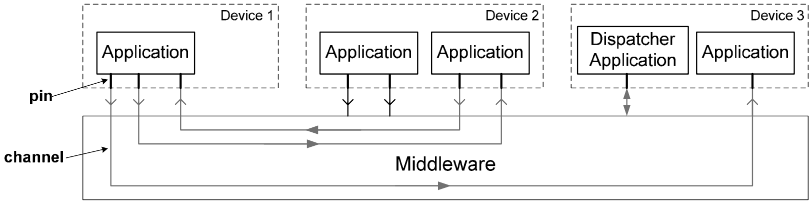


Fig. 1. MAIPAN session

3.2 Security and access control

Security and access control functions are defined and handled on device level. This means that access to services (i.e., access to application pins) are granted for devices, thus if a device gets the right to use a given service, then all applications running on this device will be able to access the service. If we assume that in the PAN there are small devices that offer one or two simple services (e.g., mouse, mp3 player), then in this case it is simpler to grant the access of services to devices instead to each application.

The dispatcher application running on a device, which plays the role of the control entity, can set up and reconfigure sessions. The control entity has to check and ask for the necessary access rights to enable the usage of a given service for the user. For instance, this main control entity can be a PDA, which has enough computing power to manage a PAN. All other devices participating in the PAN are called participants. In special cases, participants may delegate the access control rights to other devices, which will be referred to as managers.

3.3 Transferring sessions

In the PAN at least one device playing the role of the controller entity is needed. In case this device disappears all concerned sessions will be automatically torn down. To keep up such session MAIPAN offers the possibility to transfer a running session from the current control entity to another one.

For example, to make some music in a meeting room one of the users creates an mp3 playing session. This way the user's device becomes the control entity of the session. After a while, when the user wants to leave, she can transfer the session by telling to MAIPAN the identity of the new control entity, which can be for example another user's PDA.

3.4 Reconfiguring sessions

In case a participant disappears (e.g., the user leaves the room, or the device's battery is depleted), the concerned channels are automatically disconnected and the sessions have to be reconfigured. In the first step MAIPAN notifies the corresponding dispatcher(s) about the event. In the second step the dispatcher application(s) can decide which services to use instead of the disappeared ones. The dispatcher application can ask for user involvement, if there are multiple possibilities to replace the disappeared service(s), or it can decide on its own, if there is no or only one choice, or the user preferences are known. In the third step the dispatcher builds up the new channels or tears down the sessions concerned.

3.5 Architecture

Based on the concepts above we designed MAIPAN's architecture (see Figure 2). The aim of the *data plane* is to provide effective and secure data transport between applications, while the *control plane* is responsible for managing pins, channels, sessions and for handling security and access control.

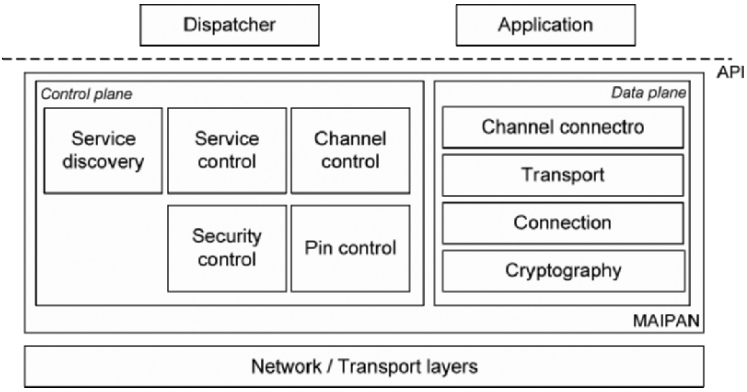


Fig. 2. Architecture of MAIPAN

3.6 Data plane

The application sends data through a pin to the middleware, where the channel connector layer redirects the data to the corresponding channel. The transport layer creates packets and provides functions such as flow control, reordering, automatic re-transmission, quality of service, etc., however these functions are not implemented, since the current version of the middleware is running over TCP/IP. The connection layer adds information to the packet, which is needed for the delivery: address of the source and destination device and the identifier of the channel. Finally, the cryptography layer calculates a message integrity check (MIC) value and encrypts the packet if necessary.

3.7 Control plane

The control plane contains the control functions which are necessary to manage the PAN. The service control part registers local services offered by the applications, handles their access rights and communicates with the service discovery protocol. The channel control creates and reconfigures sessions initiated by the dispatcher application. According to the needs of the dispatcher it asks the pin control parts of the participating devices to build up the channel between the pins. The pin control part instructs the channel connector layer to create a channel, activates the necessary transport functions for the given channel in the transport layer and sets the destination of the given channel in the connection layer. The security control part initiates and co-ordinates the authentication procedure between devices, manages the service access rights, and stores the necessary information for communication (e.g., security keys).

3.8 Implementation

MAIPAN is implemented in C on Linux 13. In this implementation the applications are connected to the middleware stack via inter-process-communication (IPC) message queues, where each pin is represented by an IPC queue. When-ever an application sends data to another application, the middleware gets the data from the IPC message queue, creates packets and sends the packet to the remote end point of the pin's channel using UDP/IPv4 for transporting. Channels are built up according to the instructions of the dispatcher application, which gets information about available services in the PAN from a simple service discovery protocol (SDP). This simple SDP is based on broadcast messages, however in theory any kind of SDP implementation could be applied.

In order to demonstrate the operation of the middleware—beside implementing the dispatcher—we created a fileserver/client application and attached an mp3 player to the system 14. This way we could set up a basic scenario, where the mp3 player could play an mp3 file located at a remote place.

4 Application programming interface

Application developers, who want to write PAN applications only and do not want to deal with PAN configuration, have to use just the service and pin registration functions of the API, which are described in the next section. Programmers, who are interested in configuring PANs and building dispatcher applications, can use all features of the API.

4.1 Service and pin registration

If an application wants to offer a service in the PAN, first it has to introduce the service to the middleware and register the input and output pins that are necessary for the use of the offered service. To do that the following message exchange is needed.

First the application sends a register service request message (Reg_Serv_Req) to the middleware. This message has to contain some service discovery protocol dependent information, which is automatically forwarded to the attached SDP. The middleware creates a new record for the service and a dedicated control channel for the application and returns an identifier, which will identify the service to the middleware in the future. Optionally the application can ask for a specific service identifier in the Reg_Serv_Req message. After this, the pins can be registered by sending a Reg_Pin_Req message for each pin. The message contains the following information: the type of the pin; whether the given pin is an input, an output or a bi-directional one; the minimum quality of service (QoS) parameters that the pin needs, such as reliability, reordering, flow control, encoding of transmitted data, etc., optionally a specific identifier for the pin.

Inside the middleware the service control forwards each pin registration request to the pin control, where the information on pins is stored. Furthermore it sets default access rights for the pins by informing the security control about the new pins. In case of successful registration, the service control acknowledges each pin registration request and returns an identifier, which will identify the pin on the given device. New pins can be registered at any time, when an application decides to do so.

Pins can be disconnected from the middleware by a Revoke_Pin_Req message, which has to contain the identifier of the pin. To disconnect services the Revoke_Serv_Req message is used, containing the identifier of the service. Revoking a service will delete all of its pins.

4.2 Gaining information about sessions, services and pins

With the Session_Info_Req message the dispatcher can query which sessions are the middleware aware of. In answer the middleware returns information on all sessions that were initiated locally and on sessions where local pins are involved. Information about services and pins can be gained with the Service_Info_Req sage. In the answer the middleware returns the locally registered services, the identifiers of the pins that belong to the services and the parameters of them. Moreover the middleware queries the SDP and forwards its answer to the application, as well.

4.3 Creating sessions

The dispatcher application can initiate the setup of sessions in the following way. First, it has to indicate that it wants to set up a session with the Create_Session_Req message. In the answer, the middleware sends an identifier for the session. After this, the dispatcher can start sending requests to interconnect pins with the Create_Channel_Req message. The control application sends as many requests as many channels it wants to establish in the given session. The message should contain the followings: the identifier of the session, the identifier of the device where the first pin is located, the identifier of the first pin, the identifier of the device where the second pin is located and the identifier of the second pin; and the quality of service (QoS) parameters that this connection needs.

Channels can be deleted with the Delete_Channel_Req message. The request contains the identifier of the session and the identifier of the channel that shall be deleted. To delete a session the Delete_Session_Req message can be used containing the identifier of the session.

4.4 Transferring sessions

The transfer of a session can be initiated by the current controller device with the Transfer_Session_Req message. The message shall contain the identifier of the session and the identifier of the new controller device. After a successful transfer the new controller will be responsible for the entire session.

4.5 Access control

The device, which is the owner of a service, can delegate its access decision right to other devices, by adding the MAIPAN ID of the device to the managers list of the given service. For this purpose the Add_Manager_Req message have to be used. As soon as the middleware receives the request to add a manager to a given service, it informs the chosen manager about the request. If the manager accepts the request, it will be added to the managers list of the service. To delete a manager the Del_Manager_Req message is used.

Each device has a white and a black lists about devices, that contain which devices are allowed and not allowed to communicate with. The white and black lists can be modified by the dispatcher application with the Set_Device_List_Req and Del_Device_List_Req messages, indicating whether the white or the black list has to be modified and which device has to be added or removed.

Also each service has a white and a black list about devices, which are allowed and which are not allowed to control the given service. The manager device stores these lists locally, so it is possible that there will be white and black lists for the same service both on the owner and on the manager, as well. The Add_Controller_Req or Del_Controller_Req messages can be sent to the middleware, indicating which service's list has to be changed and which device has to be added or removed.

4.6 Changing focus

The middleware provides facilities for connecting more pins to a given application's pin. Since data can flow only between two pins (by default towards the pin that was first connected to the source pin), similar to the X-Windows environment, the notion of "focus" arises, to determine which application receives the data flow. But unlike in X-Windows, mouse movements or the ALT+TAB key combination on the keyboard do not solve the problem, since there might be neither a mouse nor a keyboard that could be used for this purpose. In a PAN environment the concept of focus needs to be generalized and extended to handle such situations in an intuitive, ergonomic manner (e.g., placing a button on the Tetris-box, that switches between the monitors that are connected to the display output of the Tetris application). MAIPAN supports the focus change function, although, the solution of this ergonomic problem is out of scope here.

The dispatcher can initiate a focus change process with the Change_Pin_Focus_Req message. The request should contain the following information: the identifier of the session, the identifier of the device where the pin is located whose focus has to be changed, the identifier of the pin whose focus has to be changed, the identifier of the device where the pin is located which has to receive the focus and the identifier of the pin that has to receive the focus.

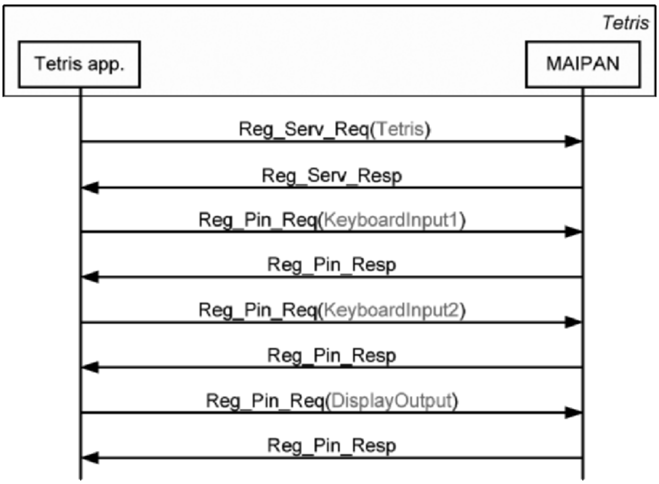


Fig. 3. Registering the Tetris

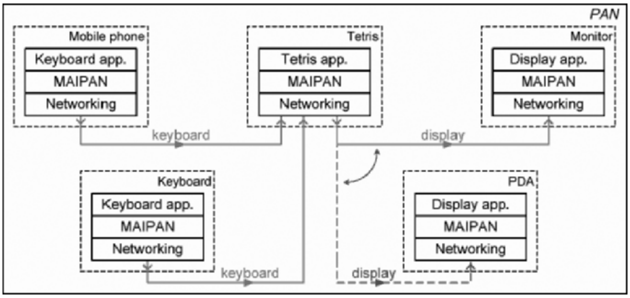


Fig. 4. Channels in the Tetris session

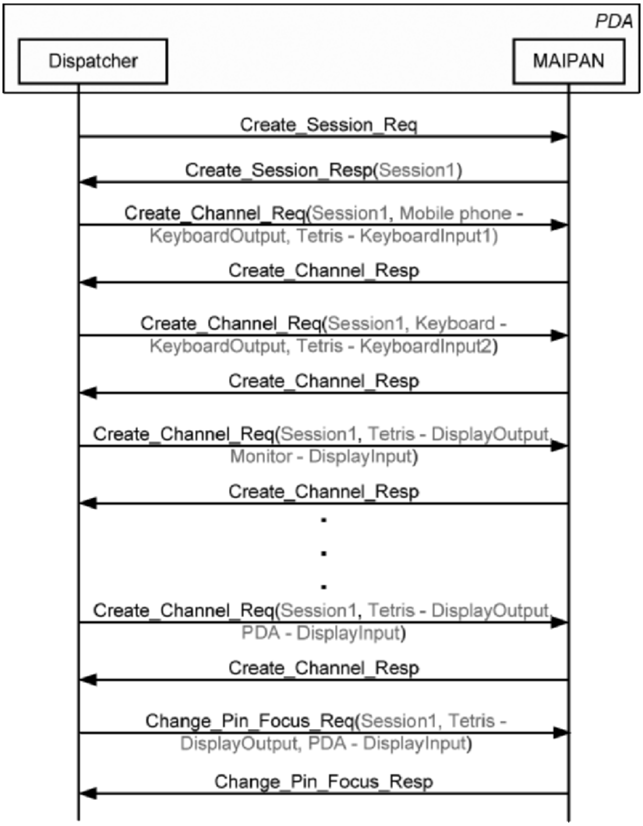


Fig. 5. Creating the Tetris session and changing focus

5 Internal operation and detailed architecture

A MAIPAN scenario can be, for example, the installation of software purchased in a MAIPAN-aware box, which is realized by adding the box to the user's PAN (see Fig. 3). Assume that the user wants to play a multi-player game with her friend in the room they are sitting. After the networking layer established the network connections, with the help of the service discovery protocol the user finds the Tetris game running on the MAIPAN-aware box. To "install" the software the user can instruct the MAIPAN dispatcher to connect one input of the Tetris-box to a PC keyboard, the other one to the keypad of her mobile phone, and the output of the Tetris-box to the large display located on the room's wall. In case the friends want to leave the room, the display output can be seamlessly redirected to the PDA's internal display—by instructing the dispatcher—, and they can continue gaming. Fig. 3 shows the channels that have to be created in the Tetris session: the arrows show the data flow between the devices, and the dashed arrow shows the configuration after the user switched from the monitor on the wall to the internal display of the PDA.

Let's see how it can be realized with the API of MAIPAN. To run the Tetris service, first the Tetris application has to register its pins to MAIPAN: two keyboard inputs and one display output is needed for the application to function properly. The message sequence of registering the Tetris service to MAIPAN can be seen in Figure 2. The mobile phone, the keyboard and the display applications has to register their services in similar way. Note that not every parameter is shown in the message sequence charts of this example.

To use the Tetris game the dispatcher application running on the PDA has to interconnect the keyboard output pin with one of the keyboard input pins of the Tetris, the mobile phone's keyboard output pin with the other keyboard input pin of the Tetris and the Tetris display output pin with the display input pin of the monitor. This message sequence can be seen in Fig. 4. From now on the game can be used by the two users.

After a while, when the users leave the room where the monitor is located, the display output of the Tetris game is redirected to the PDA's display input. This is done with the focus change function of the middleware. This message sequence is also shown in Fig. 4.

5.1 Addressing

Devices are distinguished using unique identifiers (MAIPAN ID). Devices also need a routing identifier (routing address), which is used by the routing layer to route the packets between devices (e.g., this identifier can be an IP address). The routing identifier may be changed in case network connections change, e.g., two

PANs merge or some devices appear or disappear. In contrast, unique identifiers never change.

5.2 Channel-end-point (CHEP) identifier

Channels are represented by their end-points. The identification of a channel end-point (CHEP) is twofold: it is identified by the unique identifier of the device where the end-point is located and by a CHEP identifier, which is unique on the given device. Thus channels are defined by four identifiers: two MAIPAN IDs and two CHEP IDs.

5.3 Authentication of devices

In case two devices want to communicate with each other, they have to initiate a procedure to establish a trust relationship. The exact mechanism of the authentication procedure is out of this paper's scope. However, some ideas are described in the following. The authentication procedure can be started, for example, with the following:

- The user enters a code on her own device, which is printed on the device she wants to use. With this code the devices will be able to establish a trust relationship avoiding a man-in-the-middle attack.
- The user points with her device's infra red (IrDA) port to the device she wants to use, the devices exchange some messages via IrDA and create a secure key for further communication.

After this procedure the devices can communicate in a secure way and has to know the following about each other:

- They know each other's unique identifier (MAIPAN ID).
- They are aware of each other's human-readable device name, e.g., pda111, mouse123, keyboard 456.
- They know the current routing identifier of each other (routing address).
- They have either a symmetric key or asymmetric cryptographic keys, that they use to secure their communication. Every node has a different key with every single node. The key(s) may have expiration time, in case they expire new keys have to be generated and exchanged.

The communication keys are stored in the communication secrets registry, which is handled by the cryptography layer of the data plane. Other security information (e.g., certificates) are stored in the device identities registry, which is handled by the security control part.

5.4 Access control

The service control part registers services and stores the information about them in the services registry. The service access rights are stored in the service access rights registry, which is handled by the security control part. Three levels of service access rights have been defined. The highest level is the owner, which is the device the service is running on. The owner decides whether a controller can access the given service. If the controller is welcome, then the owner puts its unique identifier into the controllers' white list of the given service. If the controller is not welcome, then its identifier is put in the controllers black list. The owner can delegate its decision right to the managers by adding the identifier of the manager to the managers list of the service. If the right is delegated, the manager stores a white and a black list per service locally, about controllers that can or cannot use the given service. The third level is the controller level. Devices that are in the controllers' white list can use the service to build up sessions; devices that are in the black list cannot use the service. These lists can be modified by the dispatcher application, and thus by the user. Managers and controllers cannot delegate their rights farther to other devices.

There are also a devices white list and a black list in the device access rights registry handled by the security control part, where the allowed and forbidden devices are listed. If a device is in the black list, then its communication establishment request is always refused. The dispatcher application running on the device can modify the content of these lists. The lists are handled by the security control part.

It depends on the operating system and on its administrator, who is allowed to run a service over the middleware. The access rights set by the operating system determine, whether a user can start a control application or a service. The middleware trusts the operating system (OS), so there is no explicit trust relationship defined between the middleware and the services running on the same OS with the middleware.

Every information on local pins—including the properties (e.g., minimum QoS requirements, direction) and the status (e.g., connected, free)—are stored in the pins registry, which is handled by the pin control part.

5.5 Session setup

A session is created, when all necessary pins are interconnected by channels. Only the control entity knows which channels form the session, thus other devices are not aware of this logical grouping of channels.

In case the channel control receives a request from a local dispatcher to connect two pins, it has to set up a corresponding channel. First, the channel control starts to create two channel end-points at the participants by sending CREATE_CHEP_REQ messages to the pin controls of the two devices (see Figure 5). Upon receiving such a request, the pin control has to check whether the controller is allowed to access the given pin. This is done by checking the access rights registry or by asking the manager of the service with an ACCESS_QUERY_REQ message.

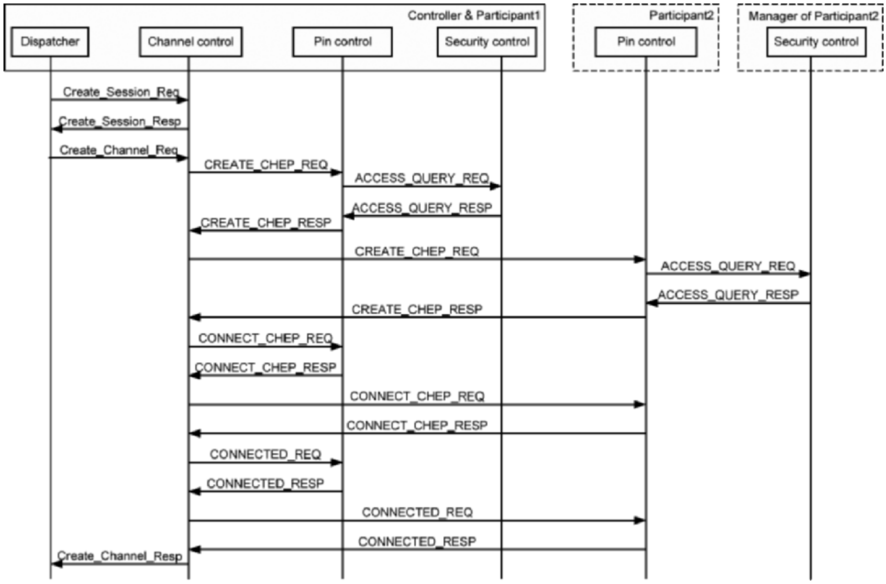


Fig. 6. Creating a channel

In case access is granted for the controller to use the given pin, the pin controller checks whether the pin is free. Connection of a new channel is only possible if the pin is not in use by another controller, or if it is in use, but the current user resigns of its usage. In the latter case an ACCESS_QUERY_REQ message is sent to the corresponding controller that uses the pin in order to determine whether it wants to resign the usage. If the controller resigns, then the new controller can use the given pin and the new channel end-point can be created.

In the second phase of setting up a channel, the two newly created channel end-points are connected to each other. This is done by sending a CONNECT_CHEP_REQ message to both participants, which contains the corresponding remote end-point of the channel and the required quality of service parameters. Thus, both participants will be aware of the channel's other end. Finally, the setup of the channel is registered in the managed channels registry of the channel control, and a CONNECTED_REQ message is sent to both participants indicating that the set-up of the channel was successful, the data transmission can be started.

In case a channel has to be connected to a pin that is already in use by the controller itself, the channel is not set up, but is registered by the initiator's channel control part as a non-active channel, i.e., a channel that does not have focus on.

MAIPAN devices, which participate in the same session, and which have not exchanged data with each other for a long time, send alive messages to indicate that they are still there. The timer of sending the alive message is reset every time, when a packet is sent to the device the timer corresponds to.

5.6 Transferring sessions

Transferring a session is done by the channel control in two steps. In the first step the new controller has to be informed about the session that has to be transferred. Thus a TRANSFER_SESSION_REQ message is sent from the old controller to the new controller, which contains the channels that belong to the session that has to be transferred.

Channel control parts cannot decide whether to accept or reject a request about transferring a session, so they have to ask the local dispatcher application. This is done by using a special notification message (EVENT).

In the second step the new controller has to rebuild the concerned channels. This is done by setting up a new session at the new controller, as shown in Figure 5. However, in this case the concerned pins are in use by the old controller, thus for each ACCESS_QUERY_REQ the old controller gives a positive answer and resigns the usage of the given pin to the new controller.

5.7 Changing focus

Although, from an application point of view, several channels can be bound to a pin at the same time, in fact at a given time only that channel is set up, which has the focus. The focus change can be initiated only by that controller whose channel currently has the focus.

The first step of the focus change is to disconnect the pin that will lose the focus. This is done by a CONNECT_CHEP_REQ message sent to the device whose pin has to be disconnected. This time the message does not contain the identifier of the CHEP to connect to, thus the CHEP at the receiver node will not have any CHEP pair and so the CHEP at the receiver will be disconnected. The second step is to create a new channel to the pin that receives the focus with CONNECT_CHEP_REQ messages.

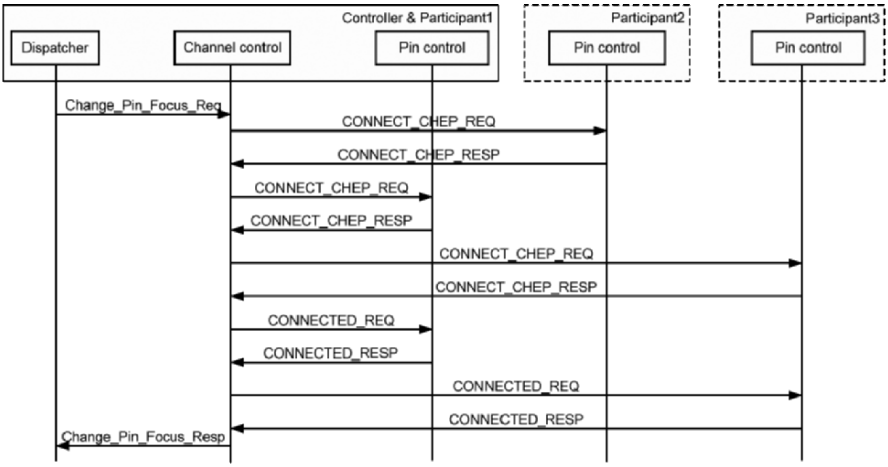


Fig. 7. Changing focus

Figure 6. shows an example, where the focus is changed at Participant1; the pin at Participant1 will be connected to a pin at Participant3 instead of its current connection to the pin at Participant2. In this case the channel between Participant1 and Participant2 will be deleted and a new channel between Participant1 and Participant3 will be created.

6 Conclusion

In this paper we described the internal operation of MAIPAN, a middleware for application interconnection in personal area networks. The essence of the middleware is to create a PAN programming platform, whereby hardware and software resources are interconnected, and the scatteredness of the PAN is hidden from the services. With MAIPAN distributed service development for mobile and nomadic environment can made easier.

MAIPAN represents a novel approach in its secure access control mechanism and the use of a central control entity. MAIPAN access control ensures 1) seamless interworking of various devices of the same user, 2) protection of one user's devices from devices of another user, 3) still enabling controlled communication and lending between devices of different users. MAIPAN manages device access and configuration via a convenient central control entity, the dispatcher. MAIPAN is also unique in enabling the change of the dispatcher role, that is, the session control rights can be transferred between devices, from the old dispatcher to a new one.

According to MAIPAN's access control scheme, access to services is granted for devices, thus if a device gets the right to use a given service, then all applications running on this device will be able to access the service. Dynamic session management handles the situation when a participant disappears. In this case MAIPAN notifies the dispatcher application, which than tries to automatically reconfigure the session by involving new participants, or if there are multiple choices and the user preferences are not known, it may ask for user involvement. MAIPAN also provides session transfer, which is useful in the situation when the device that owns a running session has to leave. In this case, if the user wants to keep up the session after she left with her device, she can instruct MAIPAN via the dispatcher application to hand over the sessions from her device to another one.

In the future we plan to enhance the current implementation, create more applications running on top of the system and extend the architecture with various new functions to improve human computer interaction.

References

1. Cecilia Mascolo, Licia Capra and Wolfgang Emmerich: “Middleware for Mobile Computing (A Survey)”, In Advanced Lectures in Networking. Editors E. Gregori, G. Anastasi, S. Basagni. Springer. LNCS 2497. 2002
2. Pope: “The CORBA Reference Guide : Understanding the Common Object Request Broker Architecture”, Addison-Wesley, Jan. 1998.
3. D. Garlan, D. P. Siewiorek, A. Smailagic, P. Steenkiste: “Aura: Toward Distraction-Free Pervasive Computing”, IEEE Pervasive Computing, 2002., http://www-2.cs.cmu.edu/ aura/
4. Gaia Project: “Active Spaces for Ubiquitous computing”, http://gaia.cs.uiuc.edu/index.html
5. “MIT Project Oxygen”, Online Documentation, http://oxygen.lcs.mit.edu/ publications/Oxygen.pdf
6. Robert Grimm, Janet Davis, Eric Lemar, Adam MacBeth, Steven Swanson, Thomas Anderson, Brian Bershad, Gaetano Borriello, Steven Gribble, and David Wetherall: “System support for pervasive applications”, ACM Transactions on Computer Systems, 22(4):421-486, November 2004.
7. CORTEX Project: “CO-operating Real-time senTient objects: architecture and EXperimental evaluation”; http://cortex.di.fc.ul.pt/index.htm
8. Barry Brumitt, Brian Meyers, John Krumm, Amanda Kern, Steven Shafer: “EasyLiving: Technologies for Intelligent Environments”, in Proc. of Handheld and Ubiquitous Computing Symposium, (Bristol, England), 2000.
9. W. K. Edwards, M. W. Newman, J. Sedivy, T. Smith: “Challenge: Recombinant Computing and the Speakeasy Approach”, MobiCom’02, September 23-28, 2002, Atlanta, Georgia, USA.
10. Jonvik, T.E., Engelstad, P.E., Thanh, D.V.: “Dynamic PAN-Based Virtual Device”, Proceedings of 2nd IASTED International Conference on

Communications, Internet and Information Technology (CIIT'2003), 17-19 Nov 2003

11. Miklós Aurél Rónai, Kristóf Fodor, Gergely Biczók, Zoltán Turányi, András Valkó: "MAIPAN: Middleware for Application Interconnection in Personal Area Networks", in proceedings of Mobiquitous 2005 conference, San Diego, CA, USA, July 17-21 2005
12. Miklós Aurél Rónai, Kristóf Fodor, Gergely Biczók, Zoltán Turányi, András Valkó: "The MAIPAN Middleware", in proceedings of Mobilfunktagung, Osnabrück, Germany, July 17-18 2006
13. Kristóf Fodor: "Implementation of a Protocol Stack for Personal Area Networks", Master's Thesis, Budapest University of Technology and Economics, June 2003
14. Balázs Kovács: "Design and Implementation of Distributed Applications in Ad Hoc Network Environment", Master's Thesis, Budapest University of Technology and Economics, May 2003

Using a SOA Paradigm to Integrate with ERP Systems

António Martins, Pedro Carrilho, Miguel Mira da Silva and Carlos Alves

Department of Computer Science, Instituto Superior Técnico, 1049-001 Lisbon, Portugal. antonio.jorge.martins@gmail.com, paqcarrilho@gmail.com, mms@tagus.ist.utl.pt, carlos.alves@outsystems.com

Abstract: Most Enterprise Resource Planner (ERP) systems already support a large part of the core organizational needs. However, other information systems need access to information stored on those ERP and, as a consequence, expensive "integrations" need to be developed between the ERP and those information systems. Sometimes, these integrations are even more expensive than the information systems themselves.
On the other hand, the Service-Oriented Architecture (SOA) paradigm has been helping businesses respond more quickly and cost effectively to information systems changes. In particular, SOA promotes reusability and interconnection of information systems – exactly what is needed for dealing with existing ERP systems.
We propose to apply the SOA paradigm to existing ERP systems so that building, changing and operating other information systems is not only faster and easier but mainly cheaper. This can be achieved by providing a tool that supports integration with ERP systems directly from other information systems without the need for ERP expertise or even write any code.
In order to demonstrate this proposal, we have implemented a tool to integrate with SAP systems from OutSystems, an Agile development framework, and evaluated this tool in a proof of concept. In the paper we present how this integration was achieved quickly and effectively without the need for any SAP expertise.

1 Introduction

As the business evolves, the applications must evolve to support them, thus requiring disposable and easy to build Enterprise Resource Planners (ERPs) and other applications that provide support for non-core processes and heterogeneous users. Evolution can only be achieved by ensuring that business processes can be changed nimbly, and this in turn is only possible when the existing applications are integrated flexibly to accommodate the required speed of change [15].

Service-oriented architecture (SOA) and Composite Applications Platforms are a solution to this problem. They allow interaction with business logic components and at the same time are deployable across a wide variety of user-defined environments, with faster deployment due to the Assembling policy [7, 8].

However, SOA environments are not usually used by ERPs. Some times this is due to the large amount of data that needs to be transferred, but mostly because ERPs owners companies prefer to provide their own solutions and modules to integrate with their ERP. SAP, for example, may require custom code that can reach up to 60% of the implementation time while on average, a costumer only uses 50% of the standard SAP code he paid for [14].

In this paper we propose to leverage an existing ERP using a SOA paradigm to integrate it with a Composite Application Platform (CAP). This way, new applications can be developed and changed quickly as also as easily integrated with the existing ERP, reducing the overall cost and project risk of ERP integration and extensibility [1, 3].

We have implemented this proposal for OutSystems HubEdition (a CAP) using SAP R/3. Developing a connector allowed the easy integration between the ERP and the CAP, enabling SAP access to OutSystems HubEdition service-oriented architecture. With this we enabled a CAP to have full integration with ERP modules and features [13].

Furthermore, we have evaluated the implementation using a real world case study (requisitions management application) and conclude that ERP can be effectively used in a SOA paradigm.

2 Service-Oriented Architecture (SOA)

To solve the interoperability problems, an effort was started to create standards for different systems to communicate in a well-known way, detached from any proprietary technology. Nowadays, Web Services are the most commonly used standard to perform this task [1, 10, 12].

Naturally, Web Services have evolved to become an implementation for an instance of a Service-Oriented Architecture (SOA), similar to the Distributed Component Object Model (DCOM) and the Common Object Request Broker Architecture (CORBA) [15].

OASIS (Organization for the Advancement of Structured Information Standards) [5] proposed a SOA reference model [15] and defined such architecture as a system of components that offer a service, which can be invoked by other components, in compliance with a service contract. According to OASIS, a Service Oriented Architecture (SOA) is a paradigm for organizing and using distributed capabilities that may be under the control of different ownership domains. In general, entities (people and organizations) create capabilities to solve or support a solution for the problems they face in the course of their business. It is natural to think of one person's needs being met by capabilities offered by someone else; or, in the world of distributed computing, one computer agent's requirements being met by a computer agent belonging to a different owner [5, 9].

2.1 Services-Oriented Development of Applications (SODA)

Gartner, on the other hand, has been working on the definition of a methodology [7, 8] capable of surpassing the obstacles of creating, changing and operating enterprise applications by using a SOA. Gartner calls this concept SODA, fitting it into the overall landscape of services-oriented architecture (SOA). SODA represents a key activity when making the transition to an SOA.

The definition of such a methodology involves the specification of the functional requirements that the tools must have, as the traditional application development tools are not suitable. This SODA methodology [7] has seven main aspects crucial for understanding the remaining of this paper:

1. **Design** - Establishment of how the requirements will be met or, in other words, definition of the application architecture. A SOA relies on loosely coupled and coarsely grained components, multi-channel access to services, process oriented design elements and early integration.
2. **Modelling** - Definition of the application's structure in some modelling language, with three distinct phases: business modelling, application modelling and technical modelling.
3. **Fabrication**- Creation of the core service components of the application and the necessary integrations. This means writing code.
4. **Assembly** - Aggregation of the service components, connection of their inputs and outputs and definition of its necessary translations. This can be done with visual editors.
5. **Orchestration** - Definition of the processes flow along the services and how information and logic will flow through a given process. This can be done with simple workflow managers.
6. **Automation** - Hiding complexity and removing the need to write code by generating it automatically to map the model on some executable component, such as .Net or Enterprise JavaBeans (EJB), enabling less tech-savvy developers to work rapidly.
7. **Variability and Rapid Change Maintenance** - Enabling changes to components that do not break the rest of the system. The variability of services should be encouraged and many small changes should be made continuously, rather than large changes sporadically. The variability of a system may be inversely proportional to its automation.

As stated, the emergence of Web services and service-oriented architecture is driving the formalization of a new style of development and tools. SODA promotes these concepts, inherent to building applications in a service-oriented world. Components become services and developers will require new integrated services environments (ISE) to build the next generation of software products.

2.2 Integrated Services Enviroment (ISE)

From these seven aspects that support the full life-cycle of applications, rises the notion of ISE that is, not only a technology stack to support service-oriented and composite applications, but also a suite of tools to sustain their development. The more it supports the seven aspects of SODA, the more complete it is. An ISE provides tools for their rapid development through composition and reusability that allow the applications to be easily changed. In addition, an ISE automates the deployment process, resulting in gains both in the development and the maintenance cycle. As a consequence, the development can be business-driven rather than technology-driven, meaning that can be focused on the business processes themselves and not on the technological details [4, 8, 10].

This paradigm also allows Agile developments [11] to be materialized in shorter cycles, allowing the feedback from real users to be incorporated in each new iteration, being immediately validated. Therefore, the application will rapidly converge to a solution which satisfies the user's needs, yielding higher success rates in projects.

2.3 Composite Applications

Composite applications are the end-products of a service-oriented architecture. They represent the business value a company derives from its SOA. Whether the composite application is designed for internal or partner use, it represents how any company can map business needs and processes to underlying information assets using SOA principles [12].

In computing, the term composite application expresses a perspective of software engineering that defines an application built by combining multiple services. A composite application consists of functionalities drawn from several different sources within the service oriented architecture (SOA). Their components may be individual web services, selected functions from other applications, or entire systems, whose outputs have been packaged as web services (often legacy systems). Composite applications often incorporate orchestration or "local" application logic to control how the composed services interact with each other to produce the new, derived functionality [3, 10].

A poorly designed composite application can endanger the performance and integrity of established production applications. A well-designed composite application can help enterprises reduce costs of IT services and improve time to market for new IT initiatives, needing careful design and planning to be built. A CA looks to the user like a regular new interactive application, yet in reality it may be only 10% new and 90% an assembly of pre-existing (purchased or in-house "legacy") components or data. The "glue" that brings a composite application together is always the integration technology. [12]

In this context, and with the growing awareness in mainstream enterprises of the opportunities offered by composite applications, users are expanding their traditional "buy vs. build" dilemma into a new "trilemma": buy, build or compose.

The compose option is not always available, and not all integration should be accomplished via only the composite application style. Yet, when the option is available, and once it is understood in its reality, composite applications emerge as a best-practice approach to modern software engineering.

3 Proposal

In this paper we propose an integration tool to use with an existing ERP and integrate it in flexible, disposable and highly configurable modules. This can be achieved using a Composite Application Platform (CAP) with SOA properties and Web-services. By creating an integration tool to link the existing parts - the ERP system and the composite platform - we have the better of both worlds.

We followed a Zero-Code approach for the integration tool [6]. This allows anyone without any SAP knowledge to create applications that reuse SAP modules without writing any code. The necessary data types are automatically converted as also as the necessary adapter code. This way the "integrator" just needs to focus on the components that he wants to integrate, and not on the specific implementation details.

Most of the above scenarios do not apply to enterprise applications. As SOA is network-centric, it requires complex monitoring and auditing of services. As services reusability and sharing are key characteristics of SOA, the number of consumer services, change management, and metering issues will be very high. These issues require a management infrastructure that may be too expensive for some projects.

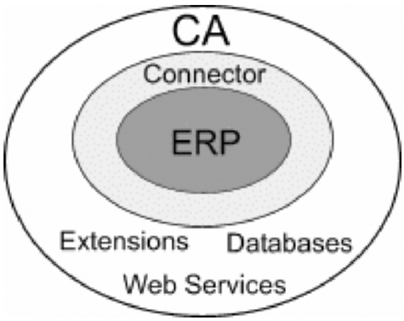


Fig. 1. ERP and Composite Application integration Architecture

This figure illustrates the essence of the ERP and Composite Application (CA) integration. Using the ERP and CA communications features, the connector can make a bridge between them. The connector is the essential integration key, because it enables ERP modules to be used by the CA accessing major SOA features. The connector maps the ERP specification to the Composite Application specification, allowing ERP access to the Composite Application. Although being the essence of the integration, the connector is also only an accelerator to the repetitive and time consuming tasks of the integration. The value of this architecture relies mostly on the CA features (mentioned before).

The presented architecture can be implemented over the majority of the existing ERPs, only requiring communication features like RPC or SOAP. In the next chapter our implementation is presented, for the mentioned architecture.

4 Implementation

Our implementation uses the SAP R/3 as ERP, the OutSystems HubEdition as the Composite Application, and the SAP Plugin for OutSystems Integration Studio as the connector.

Our connector is a plug-in for OutSystems Integration Studio, an application part of OutSystems HubEdition. OutSystems Integration Studio is an integrated visual environment where application designers can create connectors for integrating with existing enterprise systems such as SAP R/3, Navision, Siebel, etc., using agile properties (Figure 2).

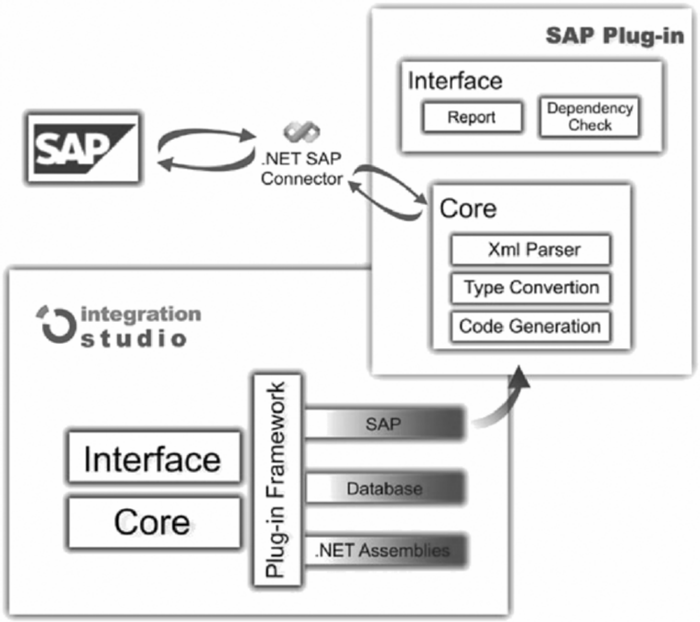


Fig. 2. SAP Plug-in architecture for OutSystems Integration Studio

Using the existing OutSystems Integration Studio plug-in framework, we created a four step wizard to guide all the integration process. We use SAP .NET Connector to connect to the SAP Server via RFC. A simple 4 step wizard (Figure 3) guides developers through the process of identifying the SAP BAPIs (Business APIs) they wish to include.

The code to invoke each SAP BAPI is automatically created by Integration Studio, as well as the mapping of the SAP data types. This way integration with SAP can be achieved in a question of minutes, removing a traditional technological roadblock.

As mentioned, the great value of this approach is not solemnly in the connector, since there are already other connectors with similar behaviors such as SAP .NET Connector or SAP Java Connector. Only by using OutSystems HubEdition features it's possible to provide fast, cheap and reusable projects with SAP Integration. All this accomplished due to the SOA approach of OutSystems HubEdition.

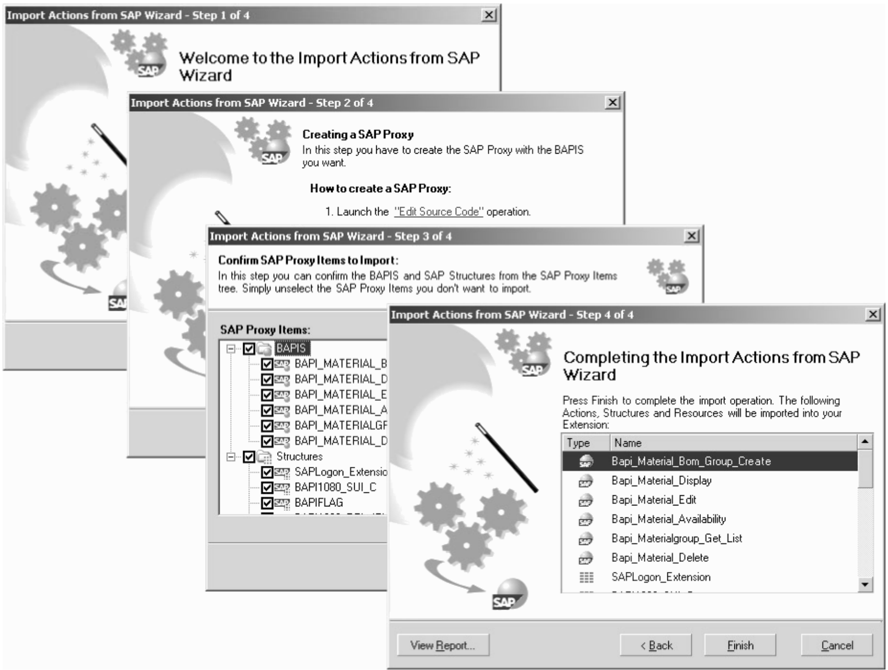


Fig. 3. SAP Plugin Wizard

5 Case Study

The best way to show the value of this approach is to provide a real world example. The solution showed in this example was implemented using a composite application.

5.1 Requisitions Management Example

The Challenge

With over 1.400 employees nationwide and a catalogue of more than 12.000 materials, the process of requisitions at ANA involved the whole company, literally generating tons of paper and requiring long manual approval cycles and error prone processes.

ANA intended to automate the requisitions process and make it available to all employees through its intranet, with the end objective of reducing overall time to request for materials, eliminate paper and reduce manual errors.

Online access was required for users to create requisitions directly from their workplace, using intuitive tools that required little training;

A simple process needed to be in place to ensure that Cost Center managers could approve requisitions easily and fast with little overhead.

The system had to integrate with SAP-MM, which was being implemented in parallel. The system had to go live in only 3 weeks, at the same time as the move into production of SAP-MM.

The Solution

- A composite application like OutSystems Hub Edition was used to create a centralized requisitions management system;
- The system has been integrated with SAP-MM to retrieve master data and trigger the creation of materials requisitions and stock reservations. The SAP integration has been achieved through an asynchronous interface to access master data periodically and a real-time interface to trigger actions directly in SAP;
- Requisitions are created by accessing the application through a regular web browser. The requisitions' front-end is available through ANA's intranet, accessible to all employees;
- When requisitions are created, Email notifications are automatically sent to the corresponding approvers. These Emails contains direct links to the online requisitions for fast access.

Once requisitions are approved, they become available to the Purchasing department, who validates them and ultimately can trigger their creation in SAP.

Company Information

ANA - Aeroportos de Portugal manages, operates and develops the Lisbon, Oporto, Faro, Ponta Delgada, Santa Maria, Horta and Flores airports, which in 2005 represented a total of 20 million passengers. Today the company has become a reference airport management group in terms of quality, profitability and ability to seize new opportunities.

6 Evaluation

Our proposal is similar to other connectors like SAP .NET Connector or SAP JCo. What these connectors don't have is the connection to a Composite Application Platform (CAP). Here relies the value to reuse and reduce integration costs, must of it due to the CAP properties, but nonetheless also provided by the connector.

A similar SOA approach also exists in SAP NetWeaver Composite Application Framework. SAP Netweaver is a technology stack comprising application server, portal and integration technology targeting the creation and extension of enterprise business processes related with ERP. Our approach is different since OutSystems Hub Edition Composite Application Platform tackles the process of rapidly creating, operating and changing user-centric enterprise applications that leverage any kind of existing enterprise systems in an agile way [11].

6.1 Proposal Advantages:

- Interact with business logic components in real-time, instead of moving data move back and forth between enterprise applications.
- Leverage Web services and the set of standards and technologies they comprise.
- Deployable across a wide variety of user-defined environments including rich, thin, mobile and portal clients.
- Create reusable business services so they can be quickly assembled and reassembled to provide new functionality for different business scenarios.
- Assembling - not coding - applications built from these business services to facilitate faster deployment.

Proposal disadvantages

The use of SOA requires additional design and development efforts and infrastructure, which translates into additional IT costs. Therefore SOA may not be a viable architecture for all cases [2].

For the following applications, SOA and Web Services may not be the recommended architecture:

- Stand-alone, non-distributed applications that do not require component or application integration; for example, a word processing application does not require request/response-based calls.
- Limited scope or short-lived applications; for example, an application that is built as an interim solution, not intended to provide functionality or reuse for future applications.
- Applications where one-way asynchronous communication is required, and where loose coupling is unnecessary and undesirable.

- A homogeneous application environment; for example, in an environment where all the applications are built using J2EE components. In this case it is not optimal to introduce XML over HTTP for inter-component communications instead of using Java remote method invocation (RMI).
- Applications that require rich, GUI-based functionality; for example, a map manipulation application with lots of geographical data manipulation is not suitable for service-based heavy data exchange.

Case Study

- The integration of the composite application like OutSystems Hub Edition with SAP-MM is exposed by Web Services that can be easily re-used in new applications without additional development costs;
- Only the Approver and Purchasing Controller profiles were formally trained. All remaining users fully mastered the requisitions front-end without any training;
- The SOA paradigm allows an agile development approach of a composite application like OutSystems Hub Edition. That way, new requirements and usability suggestions from users were quickly reflected in the requisitions system. Initially scattered organizational information is now centralized, complete and fully reusable in new applications and processes.

Integration: The system is fully integrated with SAP R/3, module MM. This integration is fully reusable by other applications without additional integration efforts, and has been achieved through the OutSystems Integration Studio SAP Plug-in.

Reach: The system is available to 1.400 users, distributed over 1.000 cost centers. Users create requisitions for up to 9 cost centers.

7. Conclusion

In this paper we have presented a proposal for a different approach to ERP integration. Through the work performed, the bases to integrate ERP components using the SOA paradigm are set.

Most organizations are service centered and not component centered. For that reason, SOA can be a major advantage since its development process and architecture are best suited for the present organizations reality.

Bearing in mind the disadvantages of this approach, we can still say that for many information systems this represents an improved solution. The connector is a valuable tool for reducing the costs and time of the integration process.

We also wanted to show that in many cases, it’s better to surround legacy systems with a modern composite application, since they will still be used, but profiting with the SOA advantages. An example was given showing the advantage of using such approach to integrate and create reusable, change-ready business services.

8 Future Work

For future work, it would be interesting to collect more benchmarking data, so we would be able to say what is faster and cheaper to integrate, with a smaller level of detail. It would also be very interesting to compare several composite applications and agile methodologies, to confront different SOA architectures. With this profiling for the integration process, we would be able to recognize specific problems and implement the correspondent concrete solutions.

References

1. Adobe (2005) Service Oriented Architecture
2. D. Garlan, R. Allen, and J. Ockerbloom (1995) Architectural mismatch or why it's hard to build systems out of existing parts. Proceedings of International Conference on Software Engineering '95, Seattle
3. Dan Woods (2003) Packaged Composite Applications - An O'Reilly Field Guide to Enterprise Software, O'Reilly
4. Daryl C. Plummer, David W. McCoy, Charles Abrams (2006) Magic Quadrant for the Integrated Service Environment Market, Gartner
5. Duane Nickull, Francis McCabe and Mathew MacKenzie (2006) RM for SOA Public Review Document, OASIS
6. Emre Kiciman, Laurence Melloul, Armando Fox (2001) Towards Zero-Code Service Composition, Stanford University
7. Gartner (2003) Next-Generation AD Tools Will Bring SOA to the Mainstream
8. Gartner (2003) The Integrated Service Environment Market Magic Quadrant
9. IBM Corporation (2005) Service Oriented Architecture: Leveraging Data Integration within a Service Oriented Architecture
10. J. Calladine (2003) Giving Legs to the Legacy – Web Services Integration Within the Enterprise. BT Technology Journal, pp 22
11. M. Aydin, F. Harmsen, Slooten van K., R. Stegwee (2005) On the Adaptation of An Agile Information Systems Development Method. Journal of Database Management, Agile Analysis, Design, and Implementation 16, pp 20-24
12. Steve Benfield and Peter Fingar (2002) Managing Web Services, SOA Webservices Journal
13. WebMethods (2005) Business Integration ROI

14. WestTrax Applications (2004) The SAP ROI Debate: What Is The Real ROI From SAP?
15. Xin Peng, Wenyun Zhao and En Ye (2004) Research on Construction of EAI-Oriented Web Service Architecture. Lecture Notes in Computer Science

SOA-MDK: Towards a Method Development Kit for Service Oriented System Development

Balbir S. Barn, Hilary Dexter[1], Samia Oussena, Dan Sparks

Thames Valley University, Wellington St, Slough, SL1 1YG, UK
{Balbir.Barn, Samia.Oussena, Dan.Sparks}@tvu.ac.uk,
Hilary.Dexter@manchester.ac.uk

Abstract: This paper argues that appropriate modeling methods for service oriented development have not matured at the same pace as the technology because the conceptual underpinning that binds methods and technology has not been sufficiently articulated and developed. The paper describes a method for service oriented development based on an adaptation and enhancement of component based techniques. The use of reference models to widen the scope of the methodology is proposed. The method is described using a complex case study from Higher Education.
Keywords: Service Oriented Architecture, Components, Methods

1 Background

There is currently a convergence to so-called Service Oriented Architecture (SOA) for application design [13] for both new developments, releasing of existing IT software assets and for integration. Enterprise systems have attained a degree of technical integration in many cases but have not yet realized the benefits of full business integration that could be gained from seamless support of the business processes. A separate move towards a stronger focus on new systems development and application integration led by business process modeling is now being enabled by the developing SOA principles [13].

Service Oriented Architecture is a disruptive technology and has the potential to provide opportunities to rethink the way systems are created and evolved. Despite this, there is still a significant lack of methodology support for SOA. A substantial section of extant literature has largely focused on the technical issues and

[1] University of Manchester

software development practices required for SOA. Little effort is discernible in the earlier parts of the software lifecycle for SOA – in particular the analysis of systems specifically designed for SOA and the subsequent identification and specification of services using model based approaches. One example where the methodology issues have been addressed to some extent is the recent work by Erl [10] where there has been an effort to recognize that service-oriented analysis is an important element in the design of effective SOA. However, here, the focus has been to derive services from a business process orchestration specification. Certainly, progression into a full service specification – using a model driven approach [11] has not been sufficiently addressed.

Thus, designers when faced with the requirements to deliver applications based on service oriented architecture do not have ready access to specific methods and techniques to support key activities such as decomposition of applications into services which can then be integrated into one or more business processes.

This paper makes a contribution to methodology research by outlining an approach to service oriented architecture by drawing on the lessons learnt and the best practices from component based practices. A key feature of the approach taken is the application of model driven architecture principles within the context of reference models for providing a methodology framework for developing systems based on SOA and Component Based Development.

2 Case study

This section provides a short description of the context of the case study for which the business process modeling and subsequent application design was performed.

The e-Framework (http://www.e-framework.org) is an initiative by the U.K's Joint Information Services Committee (JISC) and Australia's Department of Education, Science and Training (DEST) to build a common approach to Service Oriented Architectures for education and research across a number of domain areas including course validation.

The Course Validation process is one of the most important business processes within Higher Education Institutions (HEIs) and between HEIs and other institutions. New courses and the continuation of existing courses are the direct outputs of this process. Activities within the validation process are knowledge centric and collaborative. Each instance of the process is a case and will focus typically on different subject domains and therefore require different knowledge bases and experts to support the process. The end result of the process is a course specification that addresses areas such as rationale, appropriateness, justification, marketing analysis, resources required, economic viability of the courses, and detailed descriptions of the courses in terms of outcomes, aims and objectives and so on.

In this research, the purpose of the COVARM project was to define and implement the course validation business process using SOA. Systems analysis was undertaken at four institutions using a case study approach. Visual models were constructed and evaluated and an approach to synthesizing the models from each institution into a single canonical model was developed and then applied. This approach includes rules for identifying variances between processes and is described in more detail elsewhere [4]. These models were used as input to the software design and implementation stages to develop a set of software services that allowed us to automate part the business process.

3 An approach to service oriented specification and implementation

The methodology for SOA is based on two key conceptual structures or frameworks – component based development and reference models. These underpinnings are described in this section.

3.1 Building on Component based development approaches

Component based methodologies and techniques have flourished because of the exposition of a robust conceptual understanding of a component reference model. Some of the earliest work on component based development was done by Texas Instruments as part of the strategic development roadmap for the then IEF™ application development toolset [15]. The IEF was an early example of a model driven architecture in that it supported model transformation and generation. This model completeness meant that any substantial new development was required to develop a meta model to support the toolset enhancement and accompanying methodology. Thus as part of the component enhancements to the IEF a Component Based Development (CBD) conceptual model was also produced. Sterling Software continued with this meta modeling strategy when they embarked on the first pure-play component specification toolset COOL:Spex™ [1,2] based on the Catalysis Method for component based development [9].

This paper also argues that there is a conceptual similarity between component based development and service oriented approaches to systems development. This conceptual similarity has been previously identified [3, 14]. Conceptual models for CBD have been articulated in a variety of CBD methods [2, 6, 7] and an overall model is succinctly summarized in Crnkovic et al [5]. Table 1 describes concepts from components and services and provides a mapping.

Table 1. Conceptual similarity between components and services

Component Concepts	Detail	Service Concepts
Component Spec	The implementation independent view of a component	Service defined within a WSDL package
Interface	The set of related operations supported by an interface	Service in WSDL package

Interface Type Model	The set of types that represent the information that an interface needs to remember. These types are used to specify parameter data typing and form the vocabulary of any pre-post specification pairs for an operation	Data Types contained with WSDL specification
Operation	A discrete piece of functionality offered by an interface	Operation (Port Type)
Pre Post Specification Pair	Declarative specification element of an operation – used for stating the before and after state of the Interface Information Model for an operation.	None – missing concept
Packaged Component (Implementation – code, compiled component, module, deployment descriptors)	A higher level of abstraction of the implemented component / service. This will include the code that implements the component, the compiled object code, packages making up a number of compiled files; deployment descriptors and the component executable that may be deployed on a platform.	External to service specification The WSDL file contains only the location information, the protocols for deployment.

A component specification can also include a semantic specification of operations (that is the pre-post declarations of an operation), it has no equivalence in a service specification. Also a component can conceptually support more than one interface; a service has only one interface (in WSDL 1.1).

3.2 Reference Models

The second aspect of the methodology approach taken in this paper is the use of reference models to provide a contextual framework. This section presents a discussion on reference models and states what is meant by a reference model in this paper. It is argued, that there is value in using reference models for methodology development.

Inspection of a number of existing reference models such as for workflow, topic maps and sharable e-learning content [16, 17, 18] points to emerging common themes. There is an effort made to define common terms; a well-defined framework for extending aspects of the specification; attempts to define a general, overarching structure for the domain; and a focus on interoperability and standardization. These aspects are the lingua franca for a reference model. If we then consider the software engineering community as a specific example, then in addition, a reference model specifies the logical structure of the external interfaces to other systems with enough precision to be practically realizable in an efficient manner while remaining deliberately independent of any particular implementation. The codification of the interface structure will also encourage the development of software tools to enable the development of systems that conform to a particular reference model. Thus the reference model will provide a strong (perhaps enforceable) steer on how systems for a particular domain (and with specific requirements on interoperability) should be implemented.

This discussion is concluded by formalizing a notion of a reference model as follows: *"A reference model is based on a small number of unifying concepts and is an abstraction of the key concepts, their relationships, and their interfaces both to each other and to the external environment. A reference model may be used as a basis for education and for explaining standards to a non-specialist and can be viewed as a framework for comparing architectures and operations of existing and future systems"*.

Reference model concepts can be further generalised to accommodate a bigger picture of service oriented architecture.

It is this bigger picture that informs the notion of a Methodology Development Kit for SOA (SOA-MDK) that is addressed in this paper. Work on reference models for open distributed computing (RM-ODP) has already defined the needs for different viewpoints. The diagram below (Figure 1) extends that notion of viewpoints to propose that a reference model for methodology – i.e. the SOA-MDK to support SOA needs to include multiple dimensions (architectural layers) and multiple perspectives on the layers.

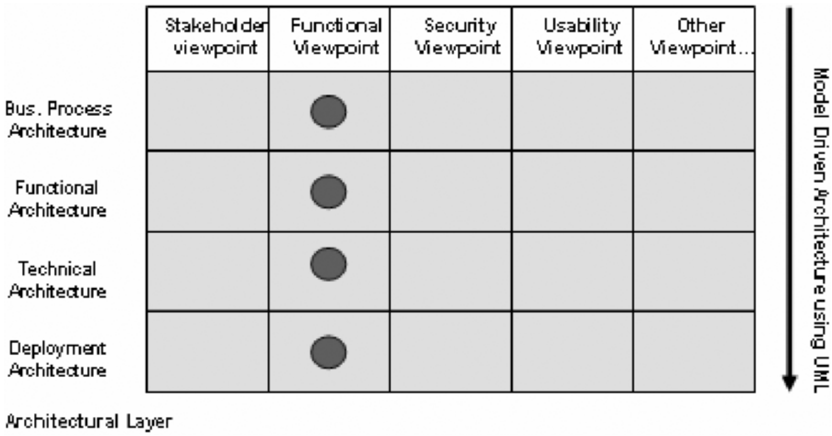


Fig. 1. Reference model framework

Thus the specification of a design of a SOA application is captured as a business process architectural description, refined into the a functional architecture expressed as a set of service specifications (the subject of the this paper) and then further refined into a technical architecture using Business Process Execution Language and finally a deployment architecture description of the location of services on specific computing hardware. Each of the architectural layers in turn, is described from specific viewpoints. The SOA-MDK viewpoint described in this paper focuses on the functional viewpoint – that is the capturing the functional requirements of the system from business process through to deployment. A key element of the approach is to focus on a model driven base utilising UML.

4 Method overview

The essence of our approach was a) to recognize and define the conceptual mappings between CBD and SOA, b) to extend and modify CBD methods to support SOA specific requirements and finally c) to ensure that a model based or model driven architectural perspective was rigorously applied from business process modeling through to service modeling using a reference modeling framework.

The key steps in the method for service oriented architecture are shown below in Fig. 2.

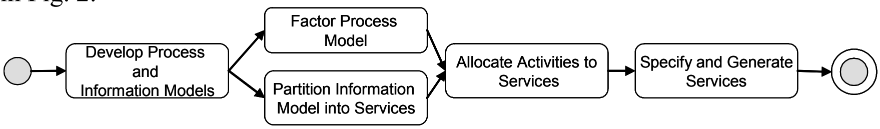


Fig. 2. Steps in the process

The activities shown are above are described briefly below.

Develop Process Models: Produce a business process model of the problem domain. Focus on activities, information flow and roles responsible for the activities.

Develop Information Models: Capture the domain concepts (types, attributes, associations and specialization relationships).Use this to define a precise vocabulary for the process models and any business rules and constraints

Factor Process Model: A process model can be come large and complex. It is useful to break down the process into a smaller set of sub-processes. We call these Event scenarios. One technique is to identify time – delay or information delay then group the activities that process a particular event as a sub-process. Rational Unified Process provides some guidelines on how to decompose processes – but there is no formality. Event Consequence Modeling is potentially a useful mechanism for factoring process models. In addition to constructing a sub-process model, the Event scenario is also supported by a textual narrative similar to a use case but with more "story". We have found that developers who were not part of the original business analysis team find these textual narratives particularly helpful.

Partition Information Model into Services: The information model is reviewed from a component perspective and a set of partitioning rules are applied [10] to create a set of components / services.

Allocate Activities to Services: Activities and sub-activities allocated to the services identified earlier. We propose using a variant of the Process/Entity Matrix from Information Engineering. Here, Activities are allocated to Services based on the information types impacted. For example, if an activity produces information which is captured by a particular information type – the service specification that contains that type will be location where the activity is allocated.

Specify service and Generate WSDL specifications: Each service comprises a number of operations. Each operation can be specified in terms of a pre and post conditions. The pre and posts are expressed in terms of types from the information model for the service. This is more or less identical to interface specification in Component Based Design. A good source of information for this and related techniques can be found in D'Souza and Wills' work on Catalysis [9]. Each service can have a WSDL (XML based) specification generated. This WSDL description is used for location and discovery and acts as implementation independent view of the service.

5 Case study example

In this section, we provide a description of how we applied the method outlined above to address the development of a number of services and their dependencies.

5.1 Develop domain models (Process Models and Information Models)

This stage entails a study of the both dynamic (process) and structural (data) information. Thus the stage requires the capture of roles/responsibilities (including teams), activities in the process, routes through the process, triggers, information consumed and produced by activities, constraints and interfaces with other information systems.

Each institution's course validation process was modeled as a UML Activity Diagram with activities (for each discernible task or action) grouped into assemblies (nested activities) corresponding to stages in the business process that were referred to as such by those responsible for setting procedure. Alongside the Activity Diagram of each institution's business process, a UML Class Diagram was created to capture the set of elements and roles in the course validation domain. This domain information model was kept at a high level of abstraction with only the key relationships between the elements included.

5.2 Factor Process Model

When a business process is a type of case process then an especially useful form of partitioning is to identify situations in the business process where there is delay in the process because there is a need for an external event to occur. When the event happens, there is consequence to the event which is set of activities that are triggered because the event occurred. There is a rich body of knowledge which supports the notion of business process understanding using this approach [8, 15].

The set of activities that are triggered can then be viewed as a sub-process of the overall business process. Such a sub-process or Event Scenario provides a better level of granularity for describing analysis scenarios for support the design and implementation stages of a software development process. An additional benefit using events to partition a business process is the direct modeling translation into BPEL specifications where there are modeling concepts for supporting events and their subsequent triggering of consequences of actions.

An Event Scenario comprising one or more activities is triggered by an event such as a time or data event. This scenario can then be analyzed by the software designer to identify operations on components. A scenario (subsetted from the overall process model) is shown below on Figure 3.

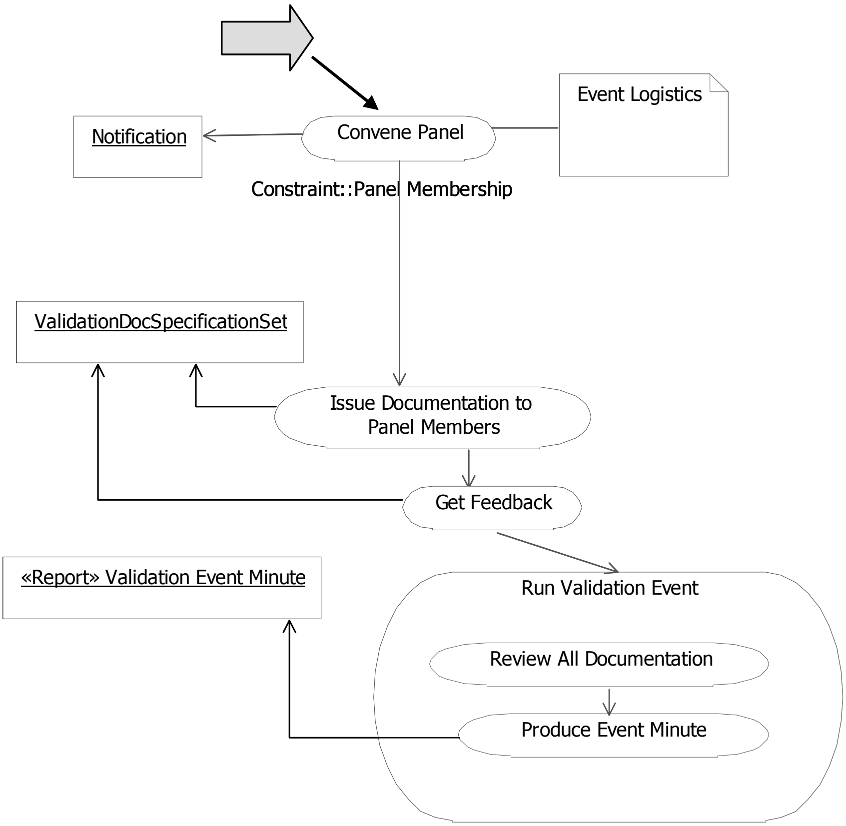


Fig. 3. Event scenario for "running an event"

5.3 Partition Information Models to Services

We propose that given the conceptual closeness between services and components, it is possible to utilize techniques from CBD. Thus, much of the service partitioning approach is obtained from the methods underpinning the component specification toolset – COOL:Spex. The approach is described in detail in [7]. In essence, the domain model or information model in our example is partitioned into "components" by firstly identifying types which are deemed to be core – that is business types or objects which are essential to the organization and then traversing associations to types that are detailing – that is providing additional details to the core type. This subsetting provides a natural component boundary. Each component identified is then allocated an interface type which will house the operations for the component.

5.4 Allocate Activities to Services

Once the components / services have been identified in this manner, the sub-process scenarios and their accompanying textual narratives are used to map activities from the process to an operation on a service. Service responsibility is based primarily on determining the types (information) being manipulated in the process and then allocating the service behaviour to the component/service that owns that type. The results of applying this technique to each of the sub-process scenarios is a set of components/services with behaviour allocated to them. A way of capturing the allocation of activities to a service is to use the so-called Process / Entity Matrix from Information Engineering [15]. A partial matrix is shown below (Table 2).

Table 2. Activity Allocations for the Event Scenario

Activity	Process Participant Information Service	Validation Document Service	Learning Product Specification Service	Event Coordination Service	Notification Service	Timetabling service
Identify appropriate panel members	■		■	■		
Notify potential panel members and recieve confimation/ unavailable responses				■	■	
Select best date and book event location				■		■
Send panel members event details and related documentation		■		■	■	

5.5 Specify Services

A service is defined by a set of operations and their parameters. These are stored in the component's information model i.e. the operations, the types and attributes of types used in the parameters of the operations. Once the component/service information model is fully specified, Rational Architect provides the generation of an XML view of the model i.e. the WSDL specification of the service.

6 Conclusion

The case study and modeling approach has demonstrated that there is sufficient conceptual equivalence between component based approaches and service oriented architecture to warrant the use of selected CBD methods. However, SOA does require an emphasis on a business process modeling and the research presented in this paper provides some enhancements to process modeling to ease the move from CBD to SOA. As we continue to develop services from new sub-process scenarios it is likely that we will refine our component partitioning strategy and the rules and good practice/guidelines for developers to support the strategy. The paper has also outlined a role for a reference model framework to provide a wider scope for methodology coverage. It is anticipated that further methodology development will address different areas of the reference framework. For example, quality of service is perceived to be an issue for service oriented architecture – so methodology to support specification of quality of service requirements from the business process architectural layer through to technical architecture i.e. the business process execution layer will be possible.

7 Acknowledgements

This work has been supported by funding from JISC – Joint Information Systems Committee http://www.jisc.ac.uk. Further details of the COVARM project are available at: http://covarm.tvu.ac.uk/covarm.

References

1. Barn BS, Brown AW, Cheesman J (1998) Methods and Tools for Component Based Development. In Tools, p. 384, Technology of Object-Oriented Languages and Systems
2. Barn BS, Brown AW (1999) Enterprise-Scale CBD: Building Complex Computer Systems From Components. In 9th International Conference on Software Technology and Engineering Practice (STEP'99), Pittsburgh, Pennsylvania, USA

3. Barn BS (2004) From Components to Web Services. Tutorial presented at ICEIS 2004, Portugal. (2004)
4. Barn BS, Dexter H, Oussena S. Petch J (2006) An Approach to Creating Reference Models for SOA from Multiple Processes. In: IADIS Conference on Applied Computing, Spain
5. Crnkovic I, Hnich B, Jonsson T, Kiziltan Z (2002) Specification, implementation, and deployment of components. Commun. ACM 45, 10, 35-40
6. Castek Ltd(2000) Component-based Development: The Concepts, Technology and Methodology. In Castek Ltd Company's white paper, available at http://www.castek.com
7. Cheesman J, Daniels J (2001) UML Components. Addison-Wesley
8. Cook S, Daniels J (1994) Designing Object Systems: Object-oriented Modelling with Syntropy. Prentice Hall
9. D'Souza DF, Wills AC (1999) Objects, Components, and Frameworks with UML: The Catalysis Approach. Object Technology Series. Addison Wesley, Reading Mass
10. Erl T (2005) Service Oriented Architecture – Concepts, Technology and Design, Prentice-Hall, USA
11. Frankel D (2004) Model Driven Architecture, OMG Press
12. Frankel D (2005) Business Process Trends. BPTrends http://www.bptrends.com/publicationfiles/07%2D05%20COL%20BP%20Platform%20%2D%20Frankel%2Epdf
13. Ort E (2005) Service-Oriented Architecture and Web Services: Concepts, Technologies, and Tools. http://java.sun.com/developer/technicalArticles/WebServices/soa2/index.html
14. Stojanovic Z, Dahanayake A, Henk S (2003) Agile Modeling and Design of Service-Oriented Component Architecture. First European Workshop on Object Orientation and Web Services. ECOOP 2003. http://www.cs.ucl.ac.uk/staff/g.piccinelli/eoows/documents/paperstojanovic.pdf
15. Texas Instruments Inc (1990) A guide to Information Engineering using the IEF™. TI Part Number: 2739756-0001.
16. Hollingsworth D (1995) The Workflow Reference Model; Workflow Management Coalition. http://www.wfmc.org/standards/docs/tc003v11.pdf
17. Durusau P, Newcomb S, Barta R. (2006) The topics map reference model. http://www.isotopicmaps.org/TMRM/TMRM-latest-clean.html
18. Advanced Distributed Learning (2004) Sharable Content Object Reference Model (SCORM) 2004 Public Draft. http://www.adlnet.gov/downloads/266.cfm

An Analysis of the Internal Structure and its Development of Virtual Organizations in the Grid

William Song* and Xiaoming Li**

*University of Durham, Durham, UK (Corresponding author)
**Peking University, Beijing, China (The paper is partly supported by the
w.w.song@durham.ac.uk

1 Introduction

For the last decade, the study on virtual organizations (VO) has gradually become an important research topic in the fields of the Grid [4]. A virtual organization can be seen as "a temporary alliance of contracted individuals or companies linked together by information and communication technologies, which assembles for the purpose of a specific business task" [6]. A virtual organization can be characterized as being a network of independent, geographically dispersed organizations. All partners in the network share the information and resources and the cooperation is based on common interest, trust, and interdependency. There is also a clear distinction between a strategic and an operational level of activity. As a characteristic of virtual organizations, interdependence is the cooperation and synergy between autonomous organizations, or between empowered departments or individuals within a single organization. Interdependence includes all types of associations: alliances, partnerships, value chains and outsourcing [7].

One of the key problems with virtual organizations in the context of the Grid is the lacking of a clear understanding of the internal structure of virtual organizations be-cause the resources are dispersed, type-vague (or we don't know their types), heterogeneous, semantically informal, and disorderly [15]. Most of the research efforts were put on semantic description and discovery of resources or services [8, 12] and se-mantic composition of services [1]. That these resources should be organized and work together to form a virtual organization and to fulfill a given task (expressed in a set of requirements) is not further studied. We think it is crucial to analyze a virtual organization, to find out its structure, and to express its behavior and working process [14].

Through analyzing the design of virtual organizations in our experiments, we see the importance of interdependence between organizations, but how to realize the interdependencies is not clear. We maintain the viewpoint that the interdependencies represent the coordination among the virtual organizations' components, which can themselves be (virtual) organizations. The components are motivated by the users' demands and requirements. We use the term task to represent a virtual organization component, or even we consider that a high level task represent a (sub-) virtual organization. Therefore, our task of the virtual organization design is to coordinate and integrate its tasks in a virtual organization.

While the tasks attempt to build up the virtual organization structure, we define services to represent the functions that are performed to meet the needs described by the tasks and to group the resources for the services. In this paper we attempt to ad-dress the problems of revealing the internal structure of VO in terms of tasks, ser-vices, and resources. As stated in [5] that the grid is considered to be a service-oriented approach, the virtual organizations should group and coordinate the grid services to flexibly serve the users' requirements.

The paper is organized as follows. In next section, we start with a simple semantic model to describe the components in VOs. We focus on the semantic description of tasks, by defining a metadata model for task description and an ontology model for tasks and services. Then we discuss the internal structure of VOs using the semantic model.

Workflow is the core process representation for VO, which bridges the tasks with services, and concerns their states and transformations. We discuss the workflow design and the VO formation in section 3. In section 4, we illustrate the task decomposition process and the workflow of VO through a demo system. Finally in section 5 we conclude the paper by discussing our research work for next steps.

2 Semantic model for VO and components

The core components in a VO are tasks and services. Tasks are considered to be the finalized representation of users' demands and requirements whereas services are the conceptual clustering of resources (being they logical or physical). Between tasks and services, a semantic matching process is, based on the properties of the tasks, to discover the most suitable services that satisfy the tasks. Once this semantic match is completed, the tasks and the services will bind together to form the VO workflow for the given objectives. In this section, we will discuss 1) what a semantic model is used to represent VO components; 2) what components to be semantically described; and 3) how to semantically represent the VO components.

Undoubtedly, semantics is extremely important in successfully building up a virtual organization and using it. In the following we address the semantic modeling problems in the users' requirements, the management of the stored resources, and the semantic uniform expressions of the virtual organization components. Firstly, user demands are often expressed in a very informal way. The proposed semantic description model should be able to describe the user demands in a formal manner with less semantic loss. Secondly, usually services and resources are already stored and de-scribed in the grid. However, their semantics is usually buried in their representation structures. The proposed semantic description model should be able to capture richer semantics out from the structures and the domain experts' expressions. Thirdly, the proposed model should be able to provide a uniform expression for them. The uniform expression requires the semantic description model to be sufficiently general to accommodate various modeling approaches of different semantic foci and precise enough to able to describe all most of the details.

2.1 Data models for tasks

As the number of components in the Grid is explosively growing, this complexity must be managed. Semantic modeling is an effective means to manage complexity in resource information construction. Semantic models help us to understand the Grid resources by simplifying some of details [3, 13]. Semantic modeling also helps us to represent main features and main structures of the Grid resources and their management and exchange. In this section, we consider two data models for tasks: Metadata model and ontology model.

2.1.1 Metadata model

In a virtual organization, tasks are the major components, which realize the end users requirements, construct workflows for the requirements, and discover and match services which then allocate and assembly the resources for the tasks. In other words, the tasks represent and formulate the users' demands and requirements at one end and deliver the basic services and consume the resources at the other end. It is obvious that the decomposition operation is a most fundamental one on the tasks. By applying the decomposition operation, the tasks form a tree structure, which supports to construct and schedule workflows for the virtual organization.

t-id:	#012
<t_1, t_2>:	t1:t2
description:	this task is to finding a suitable sort algorithm for sorting product prices.
service set:	{quick sort, bubble sort, ...}
roles:	triggers <#011, #01>, followers <#013, #121>
decomposition:	<#0121, #0122, #0123>
rules:	data type matching for <inputs, outputs>

Fig. 1. Example of the task to find a suitable sort algorithm for sorting product prices. Here the rules of the task mean that data type matching is required for the inputs and outputs.

A metadata model for a task, t, is defined to be a binary <t-id, t-attr>, where t-id is the internal representation of the task t and t-attr is a list of attributes, which are de-scribed as follows:

- <t_{begin}, t_{end}>: this is a pair of time units to indicate the beginning and the end of the task. So it delimits the life cycle of the task.
- Description: a natural language description of the task.
- Service set: a set of services from various servers. This set of services is discovered and suggested for the task to match.
- Roles: a number of actors that play various roles in performing the task, e.g., task trigger, task subject, etc.
- Rules: a set of rules, stipulating inputs to and outputs for the task to discover and match the services and resources.

In the following we illustrate the task metadata model with an example, see Fig. 1 above.

2.1.2 Ontology model

The ontology model is a special structure for a set of tasks, where only a special relationship between tasks is maintained, that is, the decomposition relationship. From the point of view of concepts, an ontology model is a tree, where all the nodes are concepts. There is one special node, called the root, which is the most general concept in the tree.

The purpose of defining an ontology model is to provide a referencing conceptual framework for a virtual organization, which we use to reason about, e.g., whether two concepts (e.g. Merge Sort and Quick Sort) belong to the same concept. The ontology model also supports tasks/service search, formulating search queries and service-task match.

Following is a widely accepted definition for the ontology. The ontology is defined to be a quintuple: concept (the concept itself), properties (all the relationships of the concept with other concepts and the attributes of the concept), axioms (rules and constraints on the concept), value domains and nominal (names or terms for the concept). However, based on our investigation on the application domain, we redefine the ontology to better accommodate the features of VO applications.

An ontology model for the tasks is defined as follows:

- Concept: a concept name is used to represent the concept of a task.
- Properties: a set of properties of the concept, e.g., execution time of the task.
- Constraints: the constraints on the concept of the tasks, e.g., disjoint (A, B), overlapping (A, B) where A and B are two concepts.
- Relationships: the semantic relations between two concepts, e.g., similar (A, B).
- Decomposition: this is a specific relationship between two tasks. This relationship results in a set of subtasks. In OWL-S [11], this relationship can be viewed to be similar to subClassOf, partOf. This forms an ontology tree for the concept of the tasks and services.

In the above example, we assume a hierarchical structure of the tasks for the virtual organization is available. This hierarchical structure is formed using domain knowledge. At the time of writing, we consult the domain experts for the decompositions patterns. The task hierarchical structured used in the above example is partially shown in Fig. 2.

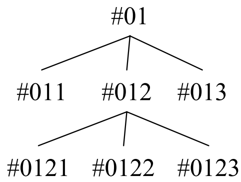


Fig. 2. In the hierarchical structure for the tasks used in the above example, the task #01 is decomposed into the subtasks #011, #012, and #013, and the task #012 is decomposed into #0121, #0122, and #0123

2.2 Services modeling

It is known to us, OWL-S provides an ontology language for web services. A description of a service covers its service profile, service process, and service grounding. The service profile provides various kinds of information about the service itself, such as contact information, service structure, inputs/outputs, and results. The service process describes the processes, which perform the tasks given by the service together with their data structures, as well as various conditions for the processes. The processes are of types: atomic and composite. The service grounding provides interfaces to the execution of services, including various parameters, input/output values, and process-related operations.

However, this kind of ontology description for web services has significant defects: Firstly, although an ontological description is provided the ontological structure for the service concepts is defined manually as there is not semantic framework is used for formally describing services. Secondly, service profile should be the place for accommodating semantic structural description for the service and the possible semantic relationships with other services. However, currently it provides only static, non-machine readable semantic information about the service. Thirdly, service process decomposition is not really the part of service ontology although it suggests that some (atomic) services may compose a composite service. Finally, atomic processes should be directly corresponding to grounding the interfaces. It would not be a se-mantic matching between the two. Rather, it should provide the names of processes, which match the names of functions (interfaces) on the ground (assuming that we had a service file library).

Therefore, we propose a semantic structure based, basic mechanism to implement a refinement structure for services ranging from high-level objective description for services to low level functional description for services. The representation model considers both the features at objectives level and concrete level. The proposed mechanism also implements a semantic matching, which bridges atomic ser-vices/processes with business and functional workflows.

A data model for a service, s, is defined to be a binary <s-id, s-attr>, where s-id is the internal representation of the task s and s-attr is a list of attributes, which are de-scribed as follows:

- Name: the name for the service.
- Input: a set of inputs and their types. The inputs to the service are from other services. The types of the inputs can help restricting the selection of services.
- Output: an output and its type. The output of the service is usually an input of the other service. The type also helps with the selection of services.
- Metadata: a set of data that describe the service, e.g., service description in natural language, service creator, service creation time, etc.
- Resource: a set of resources that is required for the service. We assume that for each service, a set of resources has already been decided, including their possible substitutes when some services fail to work.

In addition, we also assume that for a given domain a group of ontologies have been used for describing services. It is important to note the group of domain re-lated ontologies is critical for semantic matching between tasks and services.

2.3 VO workflow

In the previous definition for tasks, we have already pointed out that the decompo-sition operation on tasks will generate a tree structure for the tasks. Here we define a leaf task, denoted as t^a (atomic task), to be the task that cannot be further decom-posed. Of course, it is difficult in the current task definition to judge whether a task can be further decomposed or not. Therefore we maintain a group of task de-composition patterns, which are created through consulting with the domain experts. Using the task decomposition patterns we know whether we can further decompose a task. In order to describe a workflow in a virtual organization, we consider a setting for a task. See Fig. 3.

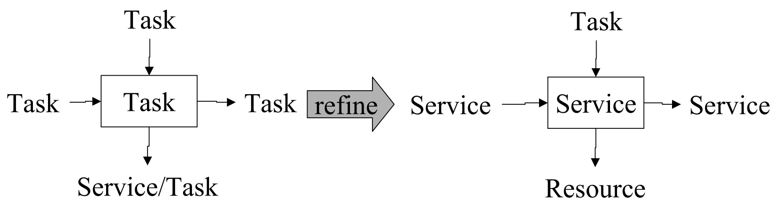


Fig. 3. The setting for a task is on the left and the setting for a service on the right

Roughly a task has two incoming arrows and two outgoing arrows. One incom-ing arrow of the task is from its parent task if the task is not a root task, to indicate that the task is a compound of its parent task. Another incoming arrow of the task is from its sibling task(s), indicating that the task is directly after its siblings (this order can be in time sequence or functional sequence). One outgoing arrow of the task is to its direct next sibling task(s). Another outgoing arrow of the task is to its compound tasks if it is not a leaf task or to its matched service(s) if it is a leaf task.

Grid services, in this circumstance, are considered to be programs (or functions), which are defined to implement the tasks and allocate the resources available in the Grid. Like the setting of a task, a service has also two incoming arrows and two out-going arrows. See Fig. 3. One incoming arrow of a service is from its task. That is, a task is associated with a set of matched services (however, these services may not be available). We stipulate that at least one service is available to realize the task. An-other incoming arrow of the service is from its preceding service(s). More than one preceding services mean that the service has more than one input parameters. One outgoing arrow of the service is to its successor services. Another outgoing arrow of the service is to the set of resources. These resources are prepared for the service. For example, the task can be "to use quick-sort method to sort the data". The service that implements the task is a quick-sort program and the resources are the functions for the program, the CPU time, the memory, the prepared data, etc.

Now, we see that there is a one-to-one mapping from the leaf tasks to the services. Using the preceding and the successor relations between the leaf tasks, we can get the following an ordered sequence:

$T_1 << T_2 << \ldots << T_n$,

where T_i, $1<i<n$, is the i^{th} leaf task in the leaf task set.

We call this sequence as the workflow of this virtual organization and the corresponding sequence of the services, $S_1 << S_2 << \ldots << S_n$, where S_i, $1<i<n$, is the i^{th} service, as an implementation of the workflow.

3 Workflow design approaches and VO formation

The formation of VO has been proposed and discussed in [4]. A more detailed description of the VO formation process was given in [10], where it is considered that a VO system only exists for a temporary time needed for satisfaction of its purpose and therefore the process of a VO is defined as consisting of needs identification, formation of VO, operations, and dissolution in the context of electronic commerce. From the context of the Grid, we consider two more steps should be added to the above VO process, i.e., searching and matching the Grid resources and integration (or composite) of services.

In the following, we discuss the steps of VO development process in detail.

The first step is users' requirements and demands solicitation and identification. A VO is created to response to end users' demands. Usually such demands provide clear goals but unclear technical descriptions. Our tasks for the first step include soliciting users' demands and requirements and document them as detailed as possible. These demands and requirements are used to guide the technical development of the VO and validate the results produced from the VO. This step is at moment done by collaboration of domain experts and VO researchers and developers.

The second step, formalizing and modeling the requirements, is to analyze and model the requirements, and form a set of general tasks. Then the tasks are further decomposed into subtasks. The process of task decomposition continues until all the leave tasks are found and a sequence of (virtually executable) tasks is formed. This step is performed by the VO developers with the help from the domain experts. It is very important that the domain knowledge and ontologies will be included in the modeling results. The forming of the task hierarchy should be conformed to the application domain ontologies, which will be used for the later matching step.

The third step is searching for and matching services and resources. According to the description of the tasks, which includes a sequence of (leave) tasks, the task hierarchy, the ontologies for the tasks, and the task requirements (specifications), the VO development environment starts to search for the set of services together with the resources that best meet the description of the tasks. The efficiency and effectiveness of the match technology depends on the quality of the semantic description for tasks, the size of the service pool, and the matching algorithm.

Forming the VO is the fourth step, where to form a workflow from the refined tasks and the match services is the main activity. The workflow here embodies the business process of the virtual organization, as well as the execution of services and resources. The resulted VO is an optimal combination of the services and resources from the service pool (we assume in this paper that a service pool exists with all kinds of services available). Therefore, one service with a longer execution time may be better than the one with a shorter execution time when meeting better the task or making a less execution time in the matched sequence of services. This depends on the policies and mechanisms for both scheduling and allocating the services that make the plan for the VO [2].

The fifth step, negotiation, integration of services and the VO execution, indicates that the above steps are iterative. Negotiation and the users' interference may be required to adjust the task decomposition and service selection so that the users' goals and demands will be better satisfied. The service integration will be achieved as long as the predefined quality for the VO is met. Then the VO is run. But not all services and resources are necessarily ready for the VO execution. The scheduling mechanism for the VO system will fetch a service when it is required.

Dissolution is the final step, in which the VO execution system (a part of the VO development environment. The other part is the construction system) will gradually release the resources when they are not required by the VO. When all the tasks complete, all the services are freed and the VO system is dissolved.

We should note that the above-discussed process is a top-down approach for the VO development based on the assumption that the service and resource pool is al-ready available. Actually, we have been developing a bottom-up approach to constitute the service pool, where the concepts and methods of software reusability and the notion of software as a service [16] have laid a foundation to construct services from resources. In addition, we assume that, during the service construction process (e.g. using OWL-S as a specification language and protégé as an implementation tool), domain knowledge and ontologies have also been recorded as a knowledge base being a part of the service pool. In other words, we make a semantic rich, “UDDI” like service registry.

In a complete process of the VO implementation, the top-down and bottom-up approaches together produce workflow for VO.

4 An example for workflow formation

The semantic description model we proposed previously aims at providing a semantic based modeling approach to discover and formalize an internal structure of a virtual organization. This approach is also supporting component analysis for the virtual organization. This analysis will further supply mechanisms for reasoning about the task hierarchy, querying the services, and discovering the resources, as well as the virtual organization workflow design. In brief, a virtual organization can be illustrated in Fig. 4.

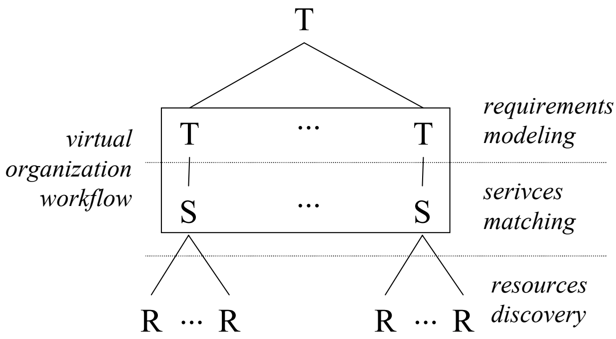


Fig. 4. The working structure of a virtual organization Here T, S, and R stand for task, service, and resource. The leaf tasks together with the services form the virtual organization workflow

Here a virtual organization starts from a high level task for general users’ requirements and demands. The task is decomposed into subtasks, which are in turn further decomposed, and hence form a task hierarchy. The leaf tasks form the workflow of the virtual organization. The task hierarchy consists of the modeling for the users’ requirements. The services which match the leaves tasks form an implementation of the workflow. The service sequence layer with the workflow represents the service de-sign and matching. The resources layer with the service sequence is the resource discovery and assembly.

In this section we use an example to illustrate our semantic description model for task decomposition and workflow. Suppose that the general task is “to sort out the product prices”, denoted T0. The first decomposition makes T0 to have the com-pound tasks T11 (“sorting”) and T12 (“product price data”). T11 is further decomposed into T211 ("sort methods") and T212 ("data"). T211 is then decomposed into T311 ("sort functions"). Now the task T311 cannot be further decomposed. It corresponds to a set of services: quick sort, bubble sort, selection sort, etc. Based on the performance required for the task, a suitable service, i.e., matched services, will be found. Part of the task hierarchy and the services are illustrated in Fig. 5.

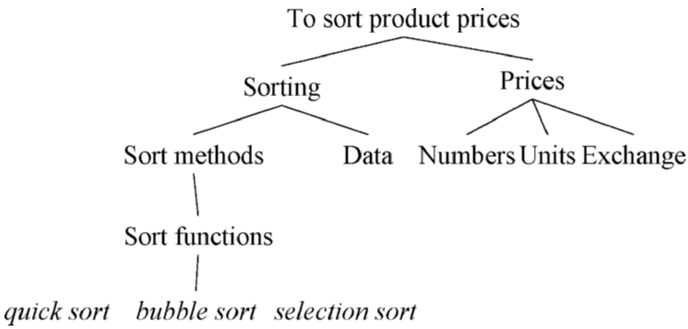


Fig. 5. The task hierarchy and the service set (in italics)

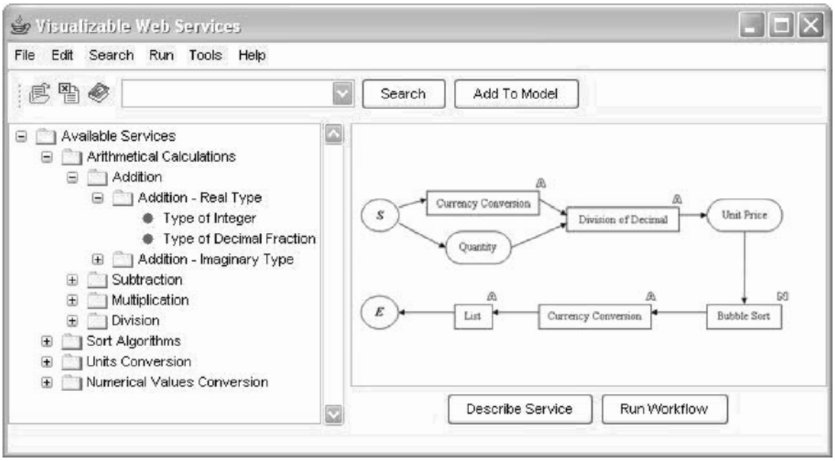


Fig. 6. A demonstration system to illustrate how the services from the task hierarchy above to form a workflow

In order to show the process of VO formation, we developed a demo system in Fig. 6. The field on the left provides a tree-structured task/service repository. All the available services in the repository, including both atomic services and composite services, are displayed there. The bar on the top is a search engine, for finding required services. A found service can be added to the modeling panel, which graphically illustrates the service composition process for a certain task. The tasks and ser-vices can be made up of atomic and non-atomic services. The panel also illustrates the flow of the task. By clicking the run button at the right bottom, the workflow is executed and shows the process of data transmission, from the starting node *S* to the ending node *E*. The description of a service can be displayed if clicking the description button.

Fig. 6 shows an example of the service of sorting the prices. The services, marked with *A* at its top right are atomic, marked by *N*, non-atomic. The function of currency conversion is to unify several kinds of currencies. After taking in the price and quantities, we can get unit prices after dividing the total price by the quantity. The unit prices will then be the input of the bubble sort. The whole flow terminates after converting the currencies back and list the result.

5 Conclusion

"Virtual organization" has become an important research issue in the areas of semantic grid computing, semantic web, e-Learning, e-Government, e-Business and so on. How to formally and semantically describe the components in a virtual organization and hence form a temporary and dynamic virtual organization is a big problem as there lacks a suitable semantic description model, which can be used to model the diverse and distributed components, including tasks, services, workflows, and resources. In this paper, we introduce a simple semantic description model for the tasks, and services, and resources in a virtual organization in order to reveal the internal structure of a virtual organization. The model includes a metadata sub-model for task property descriptions and an ontology sub-model for task conceptual descriptions. This semantic description model was proposed as the basis for the discussion of the internal structure of virtual organization and the development of virtual organization applications.

In the paper, we discuss the workflow design and the virtual organization formation in terms of these components previously described. Using a simple example in our demo system we illustrated the task-service work-flow with the internal structure for virtual organizations. Whether or not to decompose a service is still a question. As suggested in [9], to create service hierarchy will help to describe, reason, and discover, as well as maintain services. However, this depends on what kind of the internal structure we define for virtual organization. A few issues that we will explore in our future research work include:

- Formal model for virtual organization workflows. This will better support qualitative and quantitative analysis of business processes of domain applications and effective allocation and coordination of services and resources in the Grid.
- Design of a pattern template for the domain experts to formulating their domain knowledge. This is extremely important yet difficult to solve a problem as to discover the most suitable services for a domain application depends on accurate analysis and decomposition of the tasks.

References

1. Agarwal S, Handschuh S, Staab S (2005) Annotation, Composition and Invocation of Semantic Web Services, Journal of Web Semantics Vol. 2, Issue 1.
2. Alwagait E and Ghandeharizadeh S (2004) A Comparison of Alternative Web Service Allocation and Scheduling Policies, in Proceedings of IEEE SCC 2004: 319-326.
3. Conallen J (1999) Modeling Web Application Architectures with UML, Communications of the ACM, Vol. 42, No. 10, Oct. 1999.
4. De Roure D, Jennings NR, Shadbolt NR (2005) The Semantic Grid: Past, Present, and Future, Proceedings of the IEEE, Volume 93, Issue 3, March 2005, Pages 669-681, ISSN: 0018-9219
5. Goble C and De Roure D (2002b) The Semantic Web and Grid Computing, in Real World Semantic Web Applications, vol. 92, Frontiers in Artificial Intelligence and Apps., (eds. Kashyap and Shklar) IOS Press, 2002.
6. "ICENI Virtual Organisation Management", (2003) London e-Science Centre, UK, http://www.doc.ic.ac.uk/~asif/vom/index.html
7. Introna L, Moore H, Cushman M (1999) The Virtual Organisation – Technical or Social Innovation? Lessons from the Film Industry, Working Paper Series 72, Department of IS, London School of Economics and Political Science, UK.
8. Ludwig S and Reyhani S (2005) Semantic Approach to Service Discovery in a Frid Envi-ronment, *Journal of Web Semantics* Vol. 3 Issue 4.
9. Martin D, Paolucci M, McIlraith S (2004) Bringing Semantics to Web Services: The OWL-S Approach, *Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)*, July 6-9, 2004, San Diego, California, USA.
10. Oliveira et al (2003) Electronic Commerce and Virtual Organizations, http://paginas.fe.up.pt/~eol/PROJECTS/ec_vo.html
11. OWL-S (2004) http://www.daml.org/services/owl-s/1.1
12. Paolucci M, Sycara K, and Kawamuwa T (2003) Delivering Semantic Web Services, in Proceedings of WWW2003.
13. Song W (2003) Semantic Issues in the Grid computing and Web Services, in the Proceed-ings of International Conference on Management of e-Commerce and e-Government, Nanchang, China.
14. Song W and Li X (2005a) Semantic Modeling for Virtual Organization: A Case for Vir-tual Course, to appear in Advances in Information Systems Development - Bridging the Gap between Academia and Industry, (eds. Nilsson and Gustas), Springer.
15. Song W and Li X (2005b) A Conceptual Modeling Approach to Virtual Organizations in the Grid to appear in Proceedings of GCC2005 (eds. Zhuge and Fox), Springer LNCS 3795, pp. 382-393.
16. Turner M et al (2003) Turning up A Software into A Service, IEEE Computer, October 2003, pp. 38-44.

Mobile Systems Development: Exploring the Fit of XP

Ole Pedersen, Martin Lund Kristiansen, Marc Nyboe Kammersgaard, and Jens Henrik Hosbond

oleslir@gmail.com, and {martinlk, prompt, joenne}@cs.aau.dk
Department of Computer Science, Aalborg University, Denmark

Abstract: Development of mobile software is surrounded by much uncertainty. Immature software platforms on mobile clients, a highly competitive market calling for innovation, efficiency and effectiveness in the development life cycle, and lacking end-user adoption are just some of the realities facing development teams in the mobile software industry. By taking a process view on development of mobile systems we seek to explore the strengths and limitations of eXtreme Programming (XP) in the context of mobile software development. Following an experimental approach a mobile systems development project running for four months is conducted. Experiences from the project are used for analysis and discussion of the fit of XP in mobile systems development. First, requirements for mobile systems development projects are proposed. Second, these are analysed and compared to the prescribed principles suggested in XP.

In general, we find XP well-suited for mobile systems development projects. However, based on our experiences and an analytical comparison we propose the following modifications to XP: Make an essential design to avoid the worst time waste during refactoring. For faster development, reuse code components whenever possible. Test regularly on real devices, since the difference between emulators and real devices are significant. Take advantage of spikes. Do not use pair programming when spiking, and remember to write unit tests for production code that was initially created during spikes. Monitor the user during acceptance tests.
Keywords: Mobile systems development, systems development, eXtreme Programming, XP

1 Introduction

Today most people carry around mobile devices, such as mobile phones, laptops, PDAs, etc. The market for mobile devices has been growing the last many years. In 2004 alone over 670 million mobile phones were sold worldwide (Gartner

2005). With this amount of people using mobile devices, developing software for these devices becomes increasingly interesting. This article focuses solely on software for mobile phones.

Currently there are a number of things that may be said to characterize software development for mobile devices – henceforth refereed to as mobile systems development (MSD). In this paper we take a closer look at the implications of common issues reflecting development of mobile software systems. These issues are: Rapid technology change, short time-to-market, high software quality, lacking end-user adoption, and immature software tools. Through the development of mobile application steered by the principles of eXtreme Programming (XP) we explore the pros and cons of an agile and product-driven approach to MSD – in this case XP. In addition, we suggest improvements to the principles of XP, increasing the fit of XP.

The paper is structured as follows: Section 2 describes the setting of the development project and briefly sketches the software application. In Section 3 a comparative fit is made between the requirements to MSD and the prescribed principles in XP. Section 4 discusses how to improve XP development method for enhancing the support for MSD, and Section 5 ends this paper with a conclusion.

2 Research Approach

This article is based on a MSD project that adheres to the XP principles described in 'Extreme Programming Explained' by Kent Beck (Beck 1999) and 'Extreme Software Engineering' by Steinberg and Palmer (Steinberg and Palmer 2004). The development project spans a 4 month period, where we effectively spend five weeks programming the software. These five weeks are divided into three iterations. The development team is comprised of six computer science students. The students are attending their sixth and eight semester at the University of Aalborg, respectively. The customer is a sixth semester Informatics student also from the University of Aalborg.

To document the development process we use a diary in which we each day write a short description of what project activity has been performed that day, and, most importantly, what has been experienced with regards to working with the XP principles. Each entry in the diary is written in the afternoon.

Diary writing is a qualitative data collection method that is useful when evaluating projects. It enables the diary writer to reflect upon his own actions and the development method that is used during a project. It also helps a team of developers to better understand the development method, and it makes it easier to remember the development process (Jepsen, Mathiassen et al. 1989; Patton 1990; Sá 2002). For creating and managing project diaries a simple web-based tool is applied.

2.1 The Mobile System

For exploring the fit of XP in a MSD context, we set out to develop an application for streaming music to mobile devices while moving about. The client application is executed on a Java-enabled mobile phone. On application start-up, the client automatically connects to the server and is immediately hereafter able to request MP3 streams. This enables the phone to function as an MP3 player with a virtually unlimited storage capacity, even though the phone actually only has a very limited amount of physical storage. In addition, the application allows the user to purchase music while wandering about supporting any instant and spontaneous music demand the user may have.

The server runs on a stationary computer that has an arbitrary number of MP3 files stored on disk. These files are available to clients for streaming, enabling devices to connect and listen to MP3 streams through a wireless network, provided that the network is fast enough (e.g. UMTS or GPRS Edge).

Tools The server is developed using the Java 2 Standard Edition (J2SE) platform with unit tests written in JUnit. The client is developed using the Java 2 Micro Edition (J2ME) and the Wireless Toolkit. We use the Sony Ericsson J2ME SDK 2.2.0 for emulating the Sony Ericsson K750i, a mobile phone with MP3 playback support through Java method calls. The unit tests for the client are written in J2MEUnit.

2.2 The Development Process

Throughout the development process we are true to the principles of XP as suggested by Kent Beck (1999) and Steinberg and Palmer (2004). In total 13 principles are used and together they represent the development process of the project. The 13 principles are: Whole Team, Metaphor, The Planning Game, Simple Design, Small Releases, Customer Tests, Pair Programming, Test-driven Development, Design Improvement, Collective Code Ownership, Continuous Integration, Sustainable Pace, and Coding Standards. For more in-depth coverage of these principles, see 'Extreme Software Engineering' by Steinberg and Palmer (2004).

3 Analysis

In this section we suggest a list of requirements that a systems development method should support in order for it to be applicable to MSD projects, and we explain how XP should fit these requirements. Thereafter we will describe our experiences of applying XP to a simulated MSD project.

3.1 MSD Requirements

In the introduction (Section 1) we described characteristics of the mobile market. Based on these requirements we suggest the following properties that a systems development method, that is suitable for MSD projects, should support: (1) High software quality, (2) short time-to-market, (3) frequently changing user requirements, (4) frequent technology changes, and (5) lacking standardization.

(1) Upgrading the software on mobile clients costs the wireless industry $8 billion a year (Fitchard 2003). Hence, maintaining a high level of software quality is important, because it is difficult to ensure that software updates reach the mobile devices that the users carry around with them. This implies that low software quality can lead to unhappy customers, since bug fixes are unlikely to reach the end-user. Furthermore, high expenses are associated with upgrading the software that comes pre-installed on devices.

(2) A short time-to-market requirement is most likely present in any MSD project (Aarnio 2002; Hosbond and Nielsen 2005). This is due to the mobile market being largely technology-driven (Vihinen and Tuunainen 2004). The companies that produce mobile devices compete against each other, all wanting to be the first to release a new gadget to the masses.

(3) It is important to be able to handle changing user requirements. The mobile market is characterized by not yet having software that standardizes how mobile software should look and behave. In other words, the customer do not yet know exactly what they want a piece of mobile software to do, nor do they know how the software should do it (Pescatore 2004).

(4) Mobile technologies are being introduced and replaced with an incredible speed (Krogstie, Lyytinen et al. 2003; Vihinen and Tuunainen 2004; Hosbond and Nielsen 2005). Consequently, it is important to decide how to handle technology that changes frequently.

(5) There is a lack of standardization in the mobile market. The available software platforms are all different and not necessarily compatible with one another, and the software development kits are immature and poorly documented (Arnason 2004; Atkins 2004).

3.2 MSD Requirements and XP support

Comparing the mentioned requirements for MSD with the principles of XP, it becomes clear that several of the principles support the requirements. Table 1 shows these relations.

High software quality should be attainable through Test-driven Development, Acceptance Testing, Pair Programming, Design Improvements, Continuous Integration, Simple Design, and Coding Standards.

The quality of software can be perceived in several ways, e.g. the quality of the actual code, the usability of the user interface, and the users' level of satisfaction when using the software. In this section we will perceive quality of software from a professional, computer scientific point of view, meaning quality of the actual code.

Table 1. MSD requirements and supporting XP principles

MSD requirements	XP principles
(1) High software quality	Test-driven Development, Acceptance Testing, Pair Programming, Design Improvements, Continuous Integration, Simple Design, and Coding Standards
(2) Short time-to-market	Simple Design, Small Releases, Whole Team, Test-driven Development, and Continuous Integration
(3) Frequently changing user requirements	Small Releases, Whole Team, and Simple Design
(4) Rapid technology chances	Simple Design and Test-driven Development
(5) Lacking standardization	Simple Design and Test-driven Development

Test-driven Development provides better code because the code is constantly tested using unit tests. Acceptance Testing helps the programmers provide the software that the customer wants. Pair Programming should also provide better code by utilizing the positive effects of teamwork. Pair Programming has been said to be a more efficient way of writing code and that it produces software that has both a better design as well as fewer bugs (Nosek 1998). Design Improvements provide better code by improving the code design whenever possible. Simple Design and Coding Standards should reinforce the readability of the code, which leads to fewer programming mistakes and therefore better code that is more stable.

Short time-to-market is ensured by using Simple Design, Small Releases, Whole Team, Test-driven Development, and Continuous Integration. Simple Design cuts down on excessive design activity and retains only what is necessary thus giving us a shorter time-to-market. Small Releases and Whole Team should provide faster development, since the customer is involved in the development process and tests the software frequently, ensuring that the development team is not wasting precious time working in a wrong direction. Test-driven Development ensures much faster testing by automating tests, eliminating the need for spending time on repeatedly doing thorough tests of the system. Continuous Integration arranges for the system to be running at all times, regardless of its level of implemented functionality. This also leads to shorter time-to-market, since a clash, where all individually developed components of the system are put together, will never arise. Had the latter not been the case, a lot of interface conflicts would be exposed at the time everything is put together, and comprehensive adjustments of the code would be necessary.

Frequently changing user requirements are handled in XP by Small Releases, Whole Team, and Simple Design. Small Releases and Whole Team involve the customer in the development process and ensure that the software is tested by the customer during the development project. This also ensures that new user requirements, established by the customer, are identified, and Simple Design makes it easier and faster to implement these new requirements. There is no bulky design document which must be held up to date, and the system is always a minimal implementation of the current requirements. All in all, XP should make it easier to introduce changes into the system (Kusiak and He 1998).

Rapid technology changes can be handled in different ways. If you choose to stick with the same technology all the way through a development project, you need to be sure that you have chosen the right technology for your particular project. This requires that you carefully consider all possibilities before deciding on a technology to deploy and start developing. XP as is provides no support for this. If instead you choose to make your code flexible with regards to handling technology changes, XP will support you by providing Simple Design and Test-driven Development. Simple Design makes it easier to adapt to new and emerging technologies, and Test-driven Development makes it easier to test the code after it has been extended to utilize new technologies.

Lacking standardization is supported in XP by using Simple Design and Test-driven Development, since these principles make it possible to make ongoing changes during a project.

4 Discussion

The mapping of XP principles onto MSD requirements in Table 1, provide a view of the supportive properties of XP for MSD. However, experiences collected through the project diaries and day-to-day conversations called for new ways of organizing essential activities in the project. The result is set of propositions for increasing the fit of XP to MSD. The propositions are listed in Table 2.

Table 2. Propositions for further XP support in MSD projects

Proposition	Description
Essential design	Make an essential design to keep wasted time to a minimum during refactoring.
Reuse software components	Reuse object-oriented code components when possible. Remember to construct unit tests.
Develop and test on physical devices	Test regularly on physical devices. If possible, give all programming pairs access to real devices while they are developing.
Spikes	Take advantage of spikes. Do not use pair programming when spiking, and remember to construct unit tests for the code obtained through spikes.
Monitor acceptance testing	By monitoring the user while he is testing the system, it becomes clear what works and what needs improvement.

Essential design

We find that doing Simple Design and Design Improvement has some undesirable effects on our development process. If you only do Small Designs, one task at a time, the system design in entirety will be created through an incremental process, consisting of patch after patch. You will end up with a cluttered design which will require a lot of refactoring to become structured. If the whole design needs to be decomposed and refatored, it will influence nearly all components in the system. While the components are being restructured, it is not possible for other programming pairs to work. If they do, they might end up writing code for classes and methods which may not exist when the refactoring process is complete.

The only solution to this problem is for the programmers to wait for the system to be refactored, before they continue their work. This is not a very productive way of working. To avoid the worst chunks of the refactoring, we need to somehow incorporate bigger designs. We should start out by constructing an overall design. The XP way of designing, is simply not enough for MSD projects. When designing, you must think beyond the current task. This is especially true in the initial phases of a project, where the entire core structure of the code is determined. If you do not follow this guideline, you will end up having to change many lines of code repeatedly.

When the development team and the customer are gathered in order to determine user stories for the first iteration, the development team should have a conversation with the customer about which user stories he thinks are absolutely essential for the software. An essential user story is one whose absence would render the software unusable. The number of essential user stories might be bigger than the first iteration allows for, but that can be dealt with. Let the customer split the user stories into two groups: (1) *essential user stories* and (2) *optional user stories*. If some of the user stories are hard to place in either group, simply place them in group (2).

Next it is time to design a minimal system that includes the essential user stories from group (1). This design includes back-end class structure, graphical user interface, and potential protocols.

The initial design should not be implemented at once. Instead it should be well-known by all developers, and they must have it in mind when they start writing code. Every time new code is written, it must be checked for coherence with the initial design. Designs needed during later iterations are constructed ad hoc in the beginning of each iteration.

Reuse software components

Object-oriented software development has the advantage that developers write software components, which can easily be reused. This should be taken advantage of in a MSD project, where time-to-market requirements are very important. Therefore, if reuse of components is ever possible, it should be exploited. We recommend writing unit tests for components that are reused. Without them you risk introducing bugs in your code during later refactoring, which can cost time when trying to remove them.

Develop and test on physical devices

Developing software using an emulator is often cheaper and easier than developing software using a physical device. There are even situations where developers require an emulator to be present, e.g. when the software that runs on a device is being developed alongside the actual device (hardware development). When developing software using only an emulator, you expect that emulator to function exactly as the real device will do when it becomes available. Unfortunately, we have experienced that this is not always the case. Emulators and real devices often differ, making the process of getting the software up and running on a real device cumbersome and time-consuming. This is not acceptable in MSD projects, due to low time-to-market requirements. If possible, developing and testing on real devices should be done regularly during a software development process, in order to avoid major obstacles when deploying the software on the real device. Preferably, all programming pairs should have access to a real device while they are developing and testing.

Spikes

Before a project sets out, there might be technologies a development team knows nothing about that will need to be investigated. If this is not done, problems that the team is not able to handle might present themselves during development. This is of particular interest when developing software for a market that has a lot of new technologies involved, and few experts since everything is new, cutting-edge technology (Lyytinen, Rose et al. 1998)

At times, when dealing with MSD, development teams must try out several things before reaching a solution that works. For instance, it takes some research finding out if a portable device will support MP3 playback. Spikes let you do this.

When a suitable solution to a problem is found, and the development team feels confident that the new code is ready to be integrated with the rest of the system, the team should do itself the favor of writing the test cases for the code. This should be done even though it is tempting not to, now that the implementation code has already been written. Writing the test will pay off if refactoring or extension is needed later on.

Lacking standardization and rapid technology changes in MSD projects may result in situations where the development team will have lacking experience. When the team is inexperienced with something, spikes are of particular interest. Spikes are already a part of XP, but they are particular interesting in MSD because of the rapid technology changes that characterize MSD.

Pair programming has many advantages. But there are still situations where pair programming is not that useful. A pair of programmers can get stuck, not knowing how to deal with the problem at hand, and the additional team members might not be able to help them out. If this happens, a spike should be conducted.

Conducting a spike does not necessarily require the team effort of two people. It requires surfing the internet looking for implementation examples and documentation as well as searching through literature. To do this, a programmer does not need a person sitting beside him, providing support. The pair can effectively split up and conduct each their spike. Alternatively, the second programmer can team up with another team member and continue doing pair programming.

Monitor acceptance testing

Monitoring the customer while he tests the software product at each of the small releases yields great results. It allows developers to see more easily what things the end-users will experience problems with and what works well. This is especially true for the user interface. When the developers notice the customer is having problems, they must take note of it. Afterward they can discuss with the customer how the problem should be handled. This form of primitive usability test should, however, not take the place of a professionally organized usability test. The on-site customer may already be more experienced with regards to the system than the end-user will be, and he will therefore be less likely to discover as many problems as an external user will.

5 Conclusion

Based on our analysis we can conclude that XP works as an obvious development method for MSD projects. This is confirmed by our experiment of coupling XP with MSD. Furthermore, we present a series of ideas that strive to make XP an even better candidate for future MSD projects. These include: Make an essential design, reuse software components, develop and test on real devices, focus on the importance of spikes, and monitor the user during acceptance tests.

References

Arnason, A. (2004). “The wireless evolution: A journey just begun.” Rural telecommunications 23(5): 22.

Atkins, J. (2004). “New tools for new networks.” Telecommunications 38(8): 16-20.

Beck, K. (1999). Extreme Programming Explained: Embrace Change, Addison-Wesley.

Fitchard, K. (2003). “Cutting Edge: Gene Wang.” Wireless Review 20(8).

Gartner. (2005). “Gartner says strong fourth quarter sales led worldwide mobile phones sales to 30 percent growth in 2004.” from http://www.gartner.com/press_releases/asset/_121402_11.html.

Hosbond, J. H. and P. A. Nielsen (2005). Mobile Systems Development: A Literature Review. IFIP 8.2 - Designing Ubiquitous Information Environments - Socio-technical Issues and Challenges, Cleveland, Ohio, IEEE.

Jepsen, L., L. Mathiassen, et al. (1989). “Back to thinking mode: diaries for the management of information systems development projects.” Behaviour and information technology 8(3): 207-217.

Krogstie, J., K. Lyytinen, et al. (2003). Mobile information systems - research challenges of the conceptual and logical level. LNCS.

Kusiak, A. and D. He (1998). “Design for agility: A scheduling perspective.” Robotics and computer-integrated manufacturing 14: 415-427.

Lyytinen, K., G. Rose, et al. (1998). “The brave new world of development in internetwork computing architecture (interNCA): or how distributed computing platforms will change systems development.” Information systems journal 8: 241-253.

Nosek, J. (1998). “The case for collaborative programming.” Communications of the ACM 41(3): 105-108.

Patton, M. (1990). How to use qualitative methods in evaluation. Newbury Park, Sage Publications.

Pescatore, P. (2004). “Mobile consumer applications will drive data traffic and revenues over next-generation mobile networks, says IDC.” from http://www.idc.com/getdoc.jsp?containerId=pr2004_05_17_104704.

Sá, J. (2002). “Diary writing: An Interpretative research method of teaching and learning.” Educational research and evaluation 8(2): 149-168.

Steinberg, D. and D. Palmer (2004). Extreme Software Engineering, Pearson Education.

Vihinen, J. and V. Tuunainen (2004). “Identifying the limitations and capabilities of m-commerce services in gsm networks.” International journal of mobile communications 2(4): 329-324.

Aarnio, A. (2002). Adoption and use of mobile services: Empirical evidence from a Finnish survey. 35th Hawaii International Conference on Systems Sciences, Hawaii.

Implementation of Server on Grid System: A Super Computer Approach

Md. Ahsan Arefin, Md. Shiblee Sadik

Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology (BUET), Dhaka - 1000, Bangladesh.
ahsan_arefin@yahoo.co.uk, mssadik73@gmail.com

1 Introduction

The idea of metacomputing is very promising as it enables the use of a network of many independent computers as if they were one large parallel machine, or virtual supercomputer for solving large-scale problems in science, engineering, and commerce [2][6]. With the exponential growth of global computer ownership, local networks and Internet connectivity, this concept has been taken to a global level popularly called as grid computing. This, coupled with the fact that desktop PCs (personal computers) in corporate and home environments are heavily underutilized – typically only one-tenth of processing power is used – has given rise to interest in harnessing these underutilized resources (e.g., CPU cycles) of desktop PCs connected over the Internet [7][9]. This new paradigm has been dubbed as Internet computing, which is also called by several different names including enterprise/desktop grid computing, peer-to-peer (P2P) computing, and public distributed computing. There is rapidly emerging interest in grid computing from commercial enterprises. A Microsoft Windows based grid computing infrastructure will play a critical role in the industry-wide adoption of grids due to the large-scale deployment of Windows within enterprises. This enables the harnessing of the unused computational power of desktop PCs and workstations to create a virtual supercomputing resource at a fraction of the cost of traditional supercomputers.

One important application of Grid Technology can be the implementation of a Server that processes the requests of multiple Clients and perform different database applications. Normally this type of Server running on a single PC takes large execution time and caused unwanted delay to the customers (Clients). But a super computer or any other multiprocessor computers can solve this problem. Typically super computers or the multi-processor computers are not available to all and also costly. But the flavor of super computer can be achieved easily through the Grid Technology by using the underutilized resources of the available computers in the world Internet. However, there is a distinct lack of service-oriented architecture-based grid computing software in this space. To overcome this limitation, we used a Windows-based grid computing framework called "Alchemi", implemented on the Microsoft .NET Platform. [9]

2 System Architecture

The System is implemented considering the demand of faster response and processing time of the Clients or Customers for any data Query. It can be shown using the following layered architecture in the Figure 1:

A Client, who wants some data of interest, performs a database Query sending a request to the Server Application. Server handles the Query and responses the Client as early as possible by performing all the queries of the connected Clients on different executors on the Grid System. This will reduce the response time and increase throughput for the individual customer and make him happy to reuse the Server for further queries and increase its popularity.

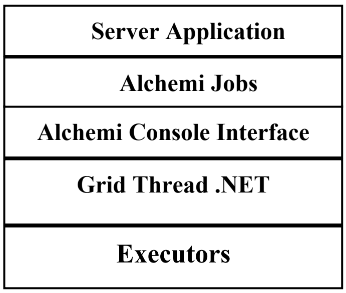


Fig. 1. Layered Architecture of Server on Grid System

2.1 Components of the System

Different types of nodes (or hosts) take part in desktop grid construction and application execution (see Figure: 2 below). Deploying a Manager node and deploying one or more Executor nodes configured to connect to the Manager construct an Alchemi desktop grid. One or more Clients can execute their applications on the cluster by connecting to the Manager through the Server. The operation of the Manager, Executor, Server, Client and Database nodes is described below.

Manager

The Manager provides services associated with managing execution of grid applications and their constituent threads. Executors register themselves with the Manager, which in turn monitors their status [6][9]. Threads received from the Server are placed in a pool and scheduled to be executed on the various available Executors. The Executors return completed threads to the Manager, which are subsequently collected by the respective Client. The Manager employs a role-based security model for authentication and authorization of secure activities. A list of permissions representing activities that need to be secured is maintained within the Manager. A list of groups (roles) is also maintained, each containing a set of permissions.

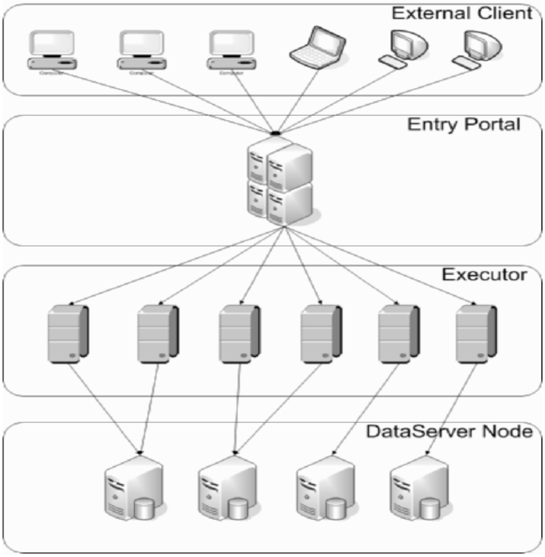


Fig. 2. Distributed Components and their Relations (Entry Portal = Manager + Server)

Executor

The Executor accepts threads from the Manager and executes them. An Executor can be configured to be dedicated, meaning the resource is centrally managed by the Manager, or non-dedicated, meaning that the resource is managed on a volunteer basis via a screen saver or explicitly by the user. In our application we used Dedicated Executors on each node. This provides two-way communication possible and exposes an interface so that the Manager may communicate with it directly. In this case, the Manager explicitly instructs the Executor to execute threads, resulting in centralized management of the resource where the Executor resides.

Server

The Server is a user written program that uses an API provided by the Alchemi developers. We used C# for our research application. The Server receives request from different Clients and performs different Database activities and some processing and calculation with the Data. The server can handle the request from multiple Clients and perform the request of all connected Clients in true parallel sense on different executors through the manager.

Client

Our Client is our User part and can be resided in any PC anywhere. We also implemented the Client using C#. In fact the Client part is totally isolated from the upper three parts, because it is not related to Alchemi rather it can be any Client written using any language whose only purpose is to connect to the Server and to send different request to the Server.

Database

The database we used here is a simple NATIONAL ADDRESS DATABASE (NAD). The format of the database is as follows (Table 1):

Table 1. The format of the database

NAD_ID	StreetNo	Street	Accuracy	DataUpID	Latitude	Longitude	StrCode
Province	PROV_ID	Town	TOWN_ID	TownCentY	TownCent X	Suburb	
SUB_ID	SuburbCentY	SuburbCentX	StreetAvgY	StreetAvgX	StreetID		
StreetName	StreetType						

The database contains about 2550000 Data. We placed the database on two (Better if more number of separate storage is used) different computers with high processing speed. Each storage conations different part of the database and responses to different queries. The Server decides which storage to use for executing the request from the Clients.

Database design is one of the trickiest parts of our System. The performance and speed of the Sever increase, if each executor uses different data storage nodes. This can happen in two ways:

- Using a replication of Database on different executor nodes.
- Another option is to use different part of the database on different executors. There parts should be independent and so are the queries.

2.2 Security of the System

Security plays a key role in an insecure environment such as the Internet. Two aspects of security addressed by Alchemi are: (a) allow users to perform authorized operations whether they are system related or resource related operations and (b) allow authorized or non-authorized users to contribute resources [4]. The problem of allowing users to only perform activities they are authorized to do is addressed using the role-based authorization model.

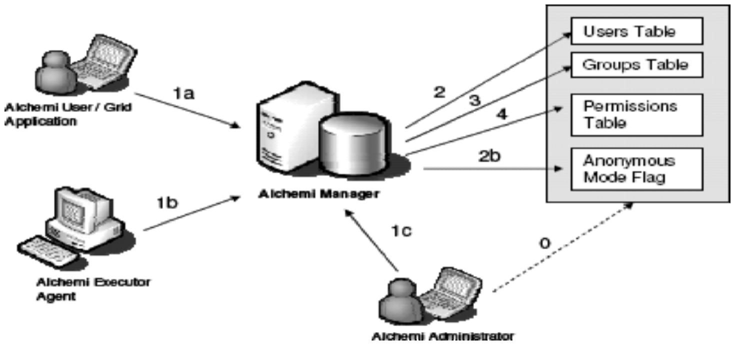


Fig. 3. Role Based Security System

Three roles are defined: User, Executors, and Administrators. We have used User role for our application

3 Implementation and Design

Let us make some review to get deeper feelings of the system. A large Database accessing and computation use large amount of CPU cycles, so now our approach is to build a powerful CPU with some idle Personal computers' CPU which are most of the time remain unused. Most of the times large numbers of Clients are connected to such a server, which access large amount of data and make same type of computation or little different type of computations those are mutually exclusive. We called this computation a job. These jobs are handled here parallely on this Grid Environment. The necessary Job model and the Thread model are described next with the sample codes.

3.1 Grid Thread Model

Minimizing the entry barrier to writing applications for a grid environment is one of Alchemi's key goals. This goal is served by an object-oriented programming environment via the Alchemi .NET API which can be used to write grid applications in any .NET-supported language.

```
public override void Start()
{
        try
        {
                m_startDateTime = DateTime.Now;
                SqlConnection thisConnection;
                SqlCommand thisCommand;
                SqlDataReader thisReader;
                thisConnection = QueryAnalyzer();
// QueryAnalyzer method creates the connection to the Appropriate Database
// for the Query
                thisConnection.Open();
                conn.Open();
                thisCommand = conn.CreateCommand();
                Console.WriteLine("Success");
                thisCommand.CommandText = m_strQuery;
                thisReader= thisCommand.ExecuteReader();
                while (thisReader.Read())
                {
                        this.m_inumber_of_row++;
                }
                Console.WriteLine("Thread {0} gets Row{1}",
                this.m_iThreadNo,this.m_inumber_of_row);
                thisReader.Close();
                thisConnection.Close();
                conn.Close();
        }
        catch (SqlException e)
        {
                Console.WriteLine(e.Message);
        }
}
```

Fig. 4. Code for Start Method (Remote Code)

"Remote code": code to be executed remotely i.e. on the grid (a grid thread and its dependencies), that is performing SQL query remotely on different Executors. And

"Local code": code to be executed locally (code responsible for creating and executing grid threads). This code is implemented here on the Server that is responsible for creating Grid Threads.

A concrete grid thread is implemented by writing a class that derives from GThread, overriding the void Start() method, and marking the Serializable class with the Serializable attribute. Code to be executed remotely is defined in the implementation of the overridden void Start() method. The sample code for this in the Figure 4:

The application itself (local code) creates instances of the custom grid thread, executes them on the grid and consumes each thread's results. It makes use of an instance of the GApplication class, which represents a grid application. The code is shown in Figure 5.

```
public void OnDataReceived(IAsyncResult asyn)
{
        SocketPacket socketData = (SocketPacket)asyn.AsyncState ;
        try
        {
                int iRx = socketData.m_currentSocket.EndReceive (asyn);
                char[] chars = new char[iRx + 1];
                System.Text.Decoder d = Sys-
                tem.Text.Encoding.UTF8.GetDecoder();
                int charLen = d.GetChars(socketData.dataBuffer,0, iRx, chars,0);
                System.String szData = new System.String(chars);
                string msg = "" + socketData.m_clientNumber + ":";
                AppendToRichEditControl(msg + szData);
                string replyMsg = "Server Reply:" + szData.ToUpper();
                byte[] byData = System.Text.Encoding.ASCII.GetBytes(replyMsg);
                Socket workerSocket = (Socket)socketData.m_currentSocket;
                workerSocket.Send(byData);
                m_richTextBoxOut.Text += "Thread No " + m_iThreadCount+ "is
                addedn";
                m_appGrid.StartThread(newAlchemiThread(szData,m_iThreadCount
                ++));
                WaitForData(socketData.m_currentSocket, socket-
                Data.m_clientNumber);
        }
        catch (ObjectDisposedException )
        {
                System.Diagnostics.Debugger.Log(0,"1","\nOnDataReceived:Socket
                has been closed\n");
        }
        catch(SocketException se)
        {
                //Error message
        }
}
```

Fig. 5. Code for Creating Thread and Application (Local Code)

Instances of the GThread-derived class are asynchronously executed on the grid by adding them to the grid application. Upon completion of each thread, a ‘thread finish’ event is fired and a method subscribing to this event can consume the thread’s results. Other events such as ‘application finish’ and ‘thread failed’ can also be subscribed to. This is shown in the Figure: 6.

3.2 Grid Job Model

Traditional grid implementations have offered a high-level, abstraction of the “virtual machine”, where the smallest unit of parallel execution is a process.

```
private void initializaApplication()
{
  try
  {       this.m_appGrid = new GApplication();
          GConnection connection = new
                GConnection("192.168.1.132",8888,"user","user");
          m_startTime = DateTime.Now;
           m_appGrid.Connection = connection;
          m_appGrid.Manifest.Add(new ModuleDepend-
          ency(typeof(AlchemiThread).Module));
           m_appGrid.ThreadFinish += new GThreadFinish(App_ThreadFinish);
           m_appGrid.ApplicationFinish += new GApplicationFin-
          ish(App_ApplicationFinish);
  }
  catch(Exception se)
  {
           MessageBox.Show(se.Message);
  }
}
private void App_ThreadFinish(GThread thread)
{
          AlchemiThread pnc = (AlchemiThread)thread;
          m_richTextBoxOut.Text += "\n"+"Total "+ pnc.m_iThreadNo +" Thread time
          ::"+(DateTime.Now - pnc.getStartTime());
}
private void App_ApplicationFinish()
{
          m_iThreadCount = 0;
          initializaApplication();
}
```

Fig. 6. Code for Finishing Thread and Application

By specifying a command, input files and output files. In Alchemi, such a work unit is termed ‘job’ with many jobs constituting a ‘task’.

```
<task>
<manifest>
          <embedded_file name="Server.exe" location=" Server.exe" />
</manifest>
<job id="0">
          <input>
                  < SQL_QUERY =" Select * from NAD where
          PROVINCE='GAUTENG' " />
          </input>
          <work run_command=" Server.exe SQL_QUERY > result1.txt" />
          <output>
                  <embedded_file name="result1.txt"/>
          </output>
</job>
</task>
```

Fig. 7. Sample XML-Based Task Representation

Tasks and their constituent jobs are represented as XML files conforming to the Alchemi task and job schemas. Figure 7 shows a sample task representation that contains one job to execute the program against two input files

Before submitting the task to the Manager, references to the ‘embedded’ files are resolved and the files themselves are embedded into the task XML file as Base64-encoded text data. When finished jobs are retrieved from the Manager, the Base64-encoded contents of the ‘embedded’ files are decoded and written to disk.

It should be noted that tasks and jobs are represented internally as grid applications and grid threads respectively.

3.3 Complete Sequence Diagram

This scheduler is called GridApplication and these threads are called Grid Threads. The sequence diagram of the total system is some thing like in the Figure 8:

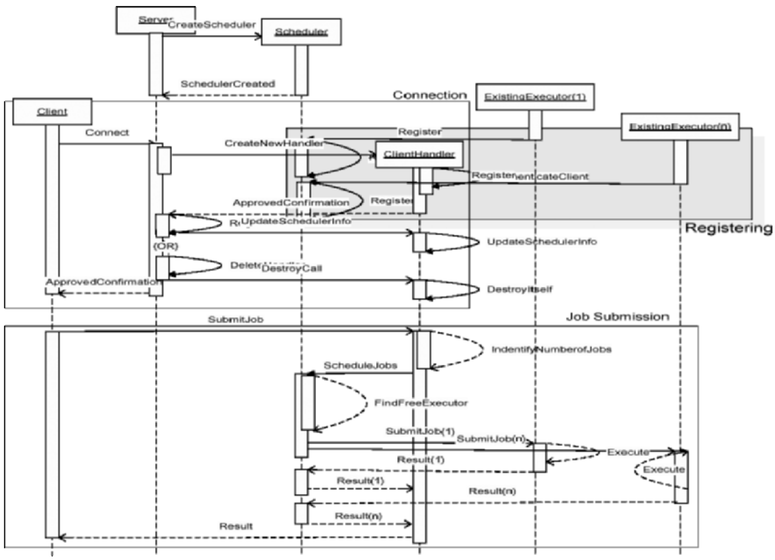


Fig. 8. Sequence Diagram (Connection and Submitting jobs)

This sequence diagram says that how these jobs are executed simultaneously on different PCs. That is any client can submit multiple jobs at a time, and the client handler will identify the No of Jobs and pass these jobs to the scheduler, and the scheduler will schedule them on different PCs. In the case of large Database access, individual executor will access the large database and compute the desired value, and the result will send back the result to the scheduler and the scheduler will send it to the ClientHandler.

4 Results and Performance

The experimental result shows a great wonder to us. We used Two PCs as Data storage nodes and compared the performance by changing the number of Executors and number of different Clients' requests. The result is shown in Table. 2

Table 2. Number of Executors Vs Execution Time

Submitted Jobs to the Server	Number of Executors	Cumulative CPU GHz	Time Min:Sec
10	1	1.694	5:14
10	2	3.189	2:28
10	3	4.692	2:20
10	4	6.368	2:07

Here the Execution time is represented in the format of Min: Sec. The Graphical plot of Table 1 (Figure 9) shows the linear improvement of the Server's Execution time with the number of Executors.

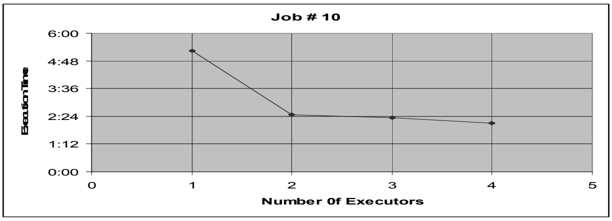


Fig. 9. Number of Executors Vs Execution Time

From this graph, we see that the performance increases much when 1 executor is replaced by the two, but the performance doesn't show a great improvement after that. This is because; we used two data storages in our experiment and each data storage can serve sequentially to the requests even though the requests are submitted to the DataServer parallely.

So the performance for three or higher executors will similarly improve much compared to the previous one if the data storage nodes can be increased with the increase of the executors

Our another experiments can prove this, shown in table: 3. Here the relation is being established between the number of Jobs and the Execution time and comparison is performed between one executor situation Vs two executors situation (see Figure 10). This really means a sense and we get the similar behavior as above if the number of executor is increased with the increases of data storage.

Table 3. Comparison for Different Number of Jobs

Submitted Jobs to the Server	Number of Executors	Cumulative CPU GHz	Time Min: Sec (Sec)
10	1	1.694	5:14 (314)
10	2	3.189	2:28 (148)
20	1	1.694	10:05 (605)
20	2	3.189	5:04 (304)
50	1	1.694	23:50 (1430)
50	2	3.189	9:45 (585)

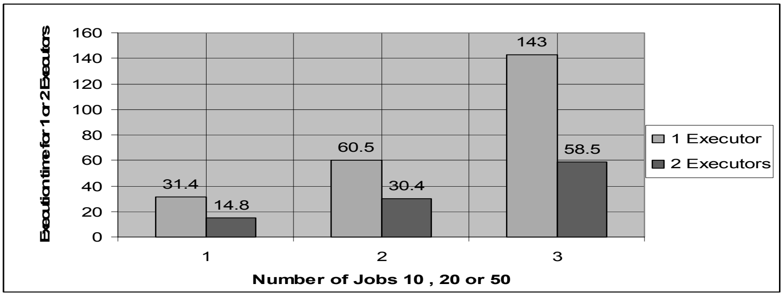


Fig. 10. Comparison for Different Number of Jobs

This can be represented mathematically as follows:
Let,

No of PC = n (all are same powerful)
No of Jobs assigned = x
And time for each job required = t
And the total time is = T
And scheduling time and networking overhead = t_s

Now, if only one CPU is available then the total time, T = x*t;

The network overhead can be neglected as today's fiber optics provides high speed and also neglecting the scheduling time, we can calculate as follows:
But, if n PCs are available then

$$\text{If } (n > x)\ T = t + t_s,\ \text{if } (t_s << t) \text{ then } T \approx t$$

$$\text{If } (n < x)\ T = (x/n)*t + t_s \text{ if } (t_s << t) \text{ then } T \approx (x/n)*t$$

Thus by increasing the number of executors with the intelligent design of database, the execution time of a Server can reduce linearly that is proportion to the number of executors connected in the Grid System.

5 Conclusion

Our Server is just an application of Grid technology that can make our life very much easy. Though we have worked with only a small server version, our latent intension is to show that this technology can be widely used to implement any complex type of server even with low overhead.

References

1. D.H. Bailey, J. Borwein, P.B. Borwein, S. Plouffe, "The quest for Pi, Math. Intelligencer ", pp. 50-57, 1997.
2. Ian Foster and Carl Kesselman, "Globus: A Metacomputing Infrastructure Toolkit", International Journal of Supercomputer Applications, 11(2): 115-128, 1997.
3. Peter Cappello, Bernd Christiansen, Mihai F. Ionescu, Michael O. Neary, Klaus E. Schauser, and Daniel Wu,Javelin "Internet-Based Parallel Computing Using Java", Proceedings of the 1997 ACM Workshop on Java for Science and Engineering Computation, June, 1997.
4. Yair Amir, Baruch Awerbuch, and Ryan S. Borgstrom, "The Java Market: Transforming the Internet into a Metacomputer", Technical Report CNDS-98-1, Johns Hopkins University, 1998.
5. Ian Foster and Carl Kesselman, "The Grid: Blueprint for a Future Computing Infrastructure", Mogan Kaufmann Publishers, USA, 1999.
6. Ian Foster, Carl Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", International Journal of Supercomputer Applications, 15(3), Sage Publications, USA, 2001.
7. Yufei Wang, Linlin Ge, Chris Rizos, Ravindra Babu, "Spatial Data Sharing On Grid", 2003.
8. Zaslavsky, I., Memon, A., Petropoulos, M., & Baru, C., "Online Querying of Heterogeneous Distributed Spatial Data on a Grid", 2003.
9. Akshay Luther, Rajkumar Buyya, Rajiv Ranjan, and Srikumar Venugopal, "Peer-to-Peer Grid Computing and a .NET-based Alchemi Framework", Conference on High Performance Computing: Paradigm and Infrastructur*e*, Laurence Yang and Minyi Guo (eds), pg. 403-429 (Chapter 21), New Jersey, USA, June 2005.

Customizing Groupware for Different Collaborative Needs

Igor Hawryszkiewycz

Department of Information systems, University of Technology, Sydney, igorh@it.uts.edu.au

Abstract: Collaboration in organization now takes place in a large variety of cultural environments that have different purposes. Purposes can include relationship management, task management, team coordination or process management. Cultures can vary from highly structured process oriented organizations to mission oriented teams in result oriented organizations. This paper provides a framework for defining such environments and choosing strategies to satisfy needs. Such strategies can range from lightweight approaches to monitored processes. It then suggests that groupware systems should be able to adapt to different environments by providing collaborative services for that environment and describes the features needed to do so.
Keywords: Collaboration, Knowledge Sharing, Groupware

1 Introduction

There are many suggestions and examples as that in (Hansen, Nohria, Tierney, 1999) for consulting organizations where collaboration improves business processes. Such improvement is especially evident where people are required to deal with increasingly complex and knowledge intensive (Grant, 1996) situations that require a quick response. In all such cases there is a need to quickly gather information and bring together people who use their knowledge to assess the situation and propose responses. The general consensus is that organizations must become agile and quickly respond to such situations in creative and innovative ways. Agility and innovation in turn require ways to support people within the organization to collaborate and use their collective knowledge to quickly provide innovative solutions.

Design of collaborative business processes must go beyond the technical flows and functions but must consider wider issues especially fitting in with culture and less defined purpose. These factors are shown in Figure 3 and include culture, especially fostering teamwork, a clear definition of goal and process, as well as providing an infrastructure to simplify collaboration. The design thus becomes multi-dimensional but emphasizes collaboration between people, which is essential to create the new knowledge needed to respond to emerging situations.

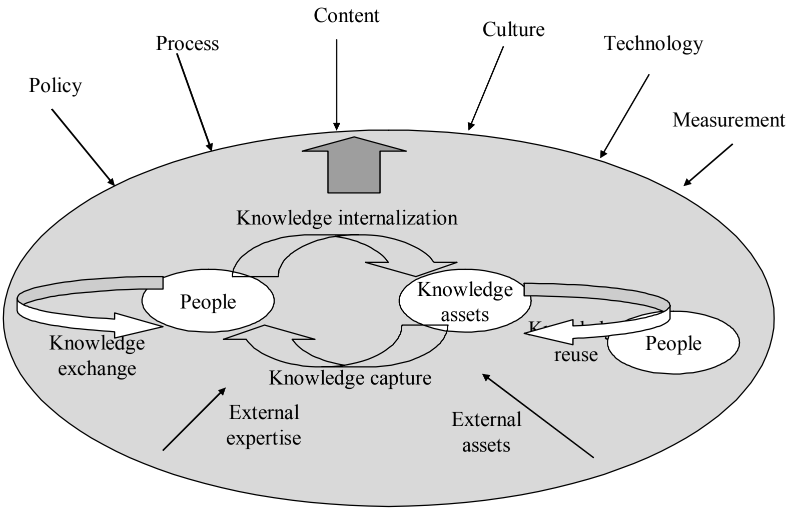


Fig. 1. Driving factors for Innovative organizations

The paper develops a design approach that enables organizations to assess the collaborative needs and ways to realize them. Collaborative technologies are important in such improvement as they support teamwork an essential component of managing any situation. This paper concentrates on collaborative technologies as they support teamwork and the exchange of knowledge and ways to integrate technologies with other driving factors shown in Figure 3. Our design approach emphasizes collaborative technology but considers the way it must be combined with the other factors shown in Figure 1. As one example are policy strategies that distinguish between personalized or codified (Hansen, Nohria, Tierney, 1999) use of information. The difference is that the former emphasizes the sharing of knowledge through personal exchanges whereas the latter sees knowledge more in codified form taking into account the work process.

2 Identifying the Issues

Design of collaborative systems is not a precise science as it is dependent on many qualitative factors including organizational culture, community practices and structure and the purpose of the collaboration. The design method must first identify and classify these factors before suggesting ways to provide technology support. The design dimensions are illustrated in Figure 2.

The major dimensions are:

Community size that ranges from open communities, small groups, through to interorganizational coordination,
The scope of the project, which may range from internal work, replying to a client request or managing an organization.
The purpose, which can range from maintaining relationships, task management, team coordination or process management,
The culture, which has been described by Hofstede (1980) who describes culture in terms of parameters such as power distance, dealing with uncertainty or individual rather collectivism,
Work practices particularly differentiating between different group structures, again distinguishing between structured, mission oriented or random groups.

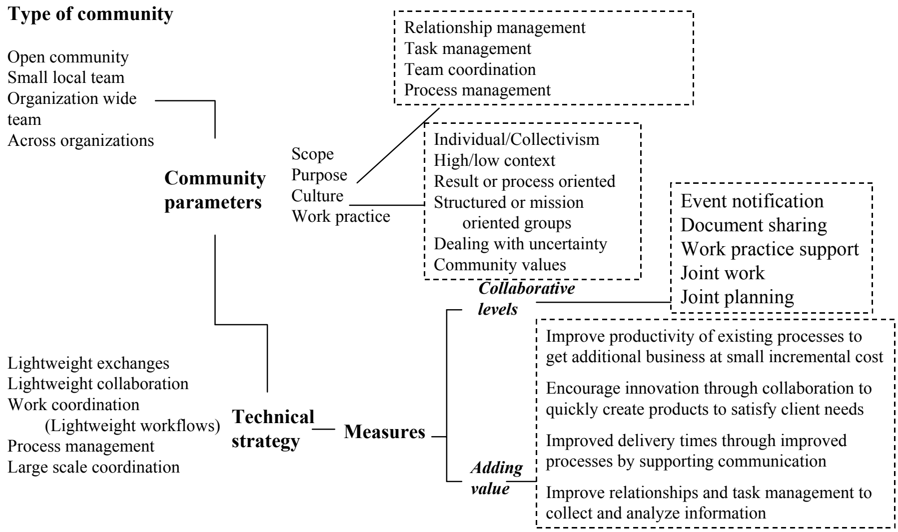


Fig. 2. The Design Dimensions

3 Defining the Technical Strategies

The strategies range from loose lightweight ones to highly structured. Lightweight in this case means little control exercised outside the people involved in the process.

Lightweight exchange primarily concerns exchange of messages between loosely connected individuals. It usually supports an environment where people stay in touch and share their responsibility but have no particular goal to achieve some outcome.

Lightweight coordination now includes the need to proceed to some outcome, although the outcome is decided as the process proceeds. Hence we now require ways to set up tasks, and assign responsibilities for them.

Work Coordination where the goal is more specific and usually requires the setting up of a plan and monitoring progress. The plan can be easily changed although the goal is usually remains the same.

Process management, where goals are now precisely defined and processes strictly followed.

The idea of lightweight exchange and collaboration was introduced by Whittaker (1999) to illustrate the kinds of technologies needed to establish and maintain productive relationships. Lightweight collaboration goes further where the exchange has some expected result as for example in (Anderson, and others, 2003) where a flexible workflow was developed for review processes in digital libraries. Lightweight in this sense means low entry barrier, flexible and web-based.

4 Choosing the Strategy

Design must to some extent satisfy the dimensions shown in Figure 2. This involves choosing the strategy, choosing the technologies to support the strategy, and providing the technologies to users through suitable interfaces. Furthermore design should be scalable and provide for evolution as often systems start with lightweight exchange but proceed to higher collaborative forms.

4.1 Relationship of Technical Strategy to Design Dimensions

Figures 3 and 4 illustrate a design decision framework. Figure 3 illustrates the relationship between the design dimensions and strategy. It shows that the choice is very subjective as it is not really possible to give a precise choice. It requires user analysis (Zhang, 2002) to determine the subjective features and then make an initial choice. As an example small groups working primarily as individuals in a large context where results are defined in general terms would probably select lightweight exchange. As the emphasis changes towards better defined results requiring collective input lightweight collaboration may be introduced. What is important is to provide technologies in ways that can support later strategic change.

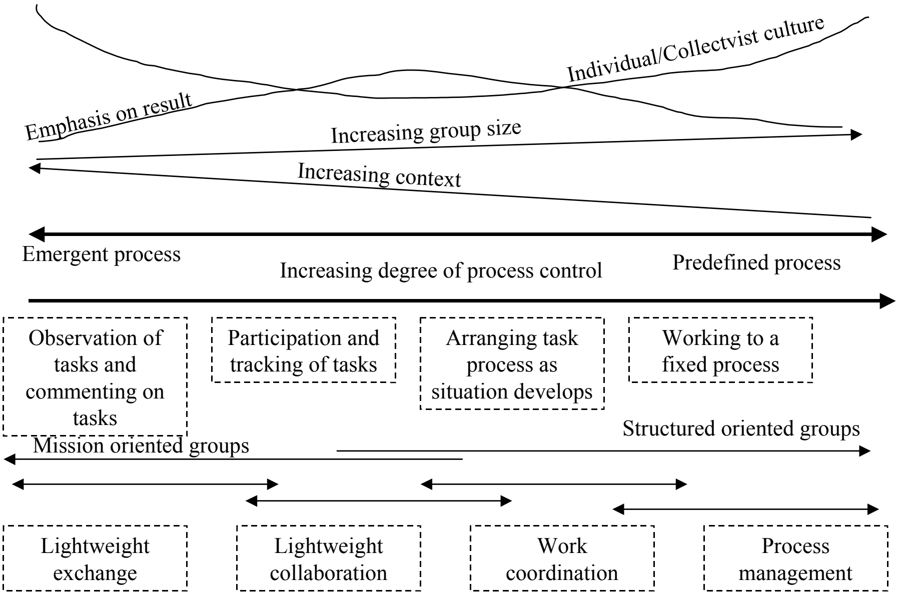


Fig. 3. Strategies and Environment

Figure 3 shows the gradual introduction of technologies that provide better ways of tracking the work.

4.2 Choosing Technologies

However adoption of collaborative technologies has not been wide and usually proceeded in an ad-hoc manner usually based on e-mail. At the same time there are few reports of effective use of leading technologies such as workspaces in business practices. Such failures often result from over reliance on technology as the major driver while ignoring its relationship to other driving factors. Most advanced technologies are used in emergency systems and in the military where technology is strongly integrated with process and peoples roles and responsibilities. Most business practices primarily using current practices such as e-mail, porportals or in some cases discussion systems and rely on individuals to adopt technologies to suit their personal preferences.

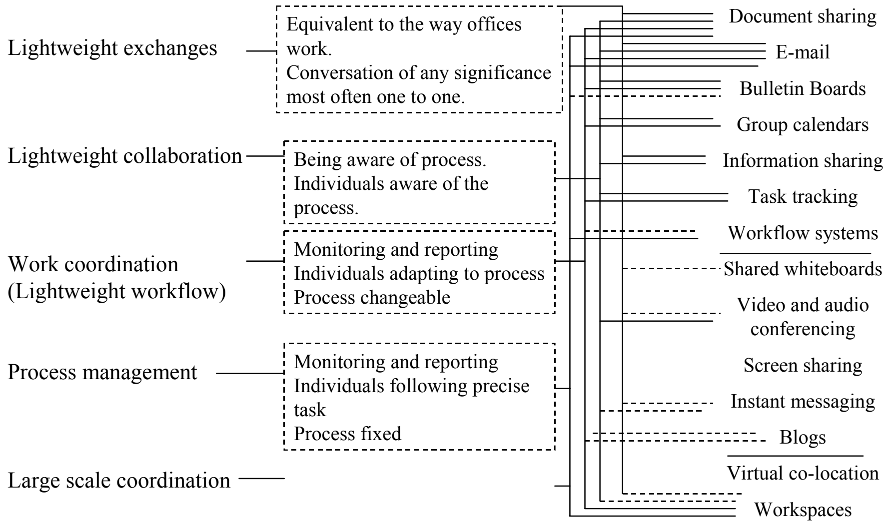


Fig. 4. Choosing Technologies to implement the Strategies

5 Measures

Measures are here divided into performance measures that can be readily measured and the less quantifiable outcome measures.

5.1 Measures of Collaboration

The collaboration levels used in this paper are shown in Table 1. These levels provide a way to gradually increase levels of sophistication as the interaction between people increases with each level. Table 1 describes each level and its characteristic, the technology needed as well as the knowledge needed to realize the level. The levels are:

- Event notification, where roles are informed of any changes that effect the roles.
- Document sharing, where documents are distributed between responsible roles,
- Work process support, which often defines monitoring levels of activity and sending reminders to collaborators,
- Joint work, where users work together in a synchronous manner, and
- Joint goal setting, where people jointly decide how they will work together.

Usually collaboration starts with awareness, which simply concerns notifying people of changes that can impact on their work. Then document sharing is added to ensure that people are provided with information needed to carry out their responsibilities. Subsequent levels are more complex as they require more intense interaction to coordinate activities. Work process support, requires a precise definition of the way a collaborative process takes place. It includes the definition of responsibilities of identified roles. For example the process may define a way to develop a response to a customer request. The specific rules may define the expertise needed to define a solution, the risk assessment, budgetary evaluation, legal aspects and so on. The structure of documents may also be specifically defined. This process is clearly understood and followed. Joint work goes further in that many of the activities may be carried out synchronously thus reducing completion time. Joint goal setting is where involved units together plan and agree on their work processes. This level often requires support for asynchronous work as goal setting often includes resolving many imprecisely defined alternatives.

Table 1. Levels of Collaboration

Level	Characteristics of collaboration levels	Technologies needed	Knowledge requirements
Collaboration level 1 Awareness	Informing people about events related to their roles. Presenting the functional situation globally.	e-mail alerts. SMS messages. Visual displays. Information portals.	People responsibilities in the organization and their expertise and availability to direct alerts.
Collaboration level 2 Document Sharing	Sharing explicit information. Presenting to roles responsible for functional units.	e-mail, web portals	Information requirements of different roles.
Collaboration level 3 Work process support	Explicit definition of work activities and responsibilities. Definition of relationships between tasks. Group meeting.	e-mail Coordination and Workflow tools, Blogs, Video displays, Calendar systems.	Optimum team structures for identified situations. Location of experts. Ways to assign responsibilities.
Collaboration level 4 Joint work	Jointly create and develop artifacts.	Shared white-boards. Workspaces.	Responsibilities of business units.
Collaboration level 5 Joint goal setting	Developing shared plans. Devise coordination strategies between functional units.	Synchronous video communication.	Organizational strategy and mission.

The choice depends on the where the best benefits are identified. For example collaboration is essential in supply chains. The goals here depend on the situation. Rodin (2006) for example emphasizes the need to maintain awareness between organizations in a supply chain to reduce delivery times in situations where customers change their requirements. This requires collaboration capability 1 across all steps of the chain to ensure that all steps are notified if there are unanticipated delays in one step. They can then take their own corrective actions. Alternately in more critical situation collaboration capability 2 may be needed to help users to quickly adjust activities across the entire supply chain.

6 Groupware support

Figure 4 provides the guidelines for the design of workspace systems. What one is looking for is ways to start with lightweight exchanges and gradually change the strategy. Figure 5 shows such evolution. It uses our LiveNet system to evolve collaboration to illustrate the development of interfaces to respond to a new competitor.

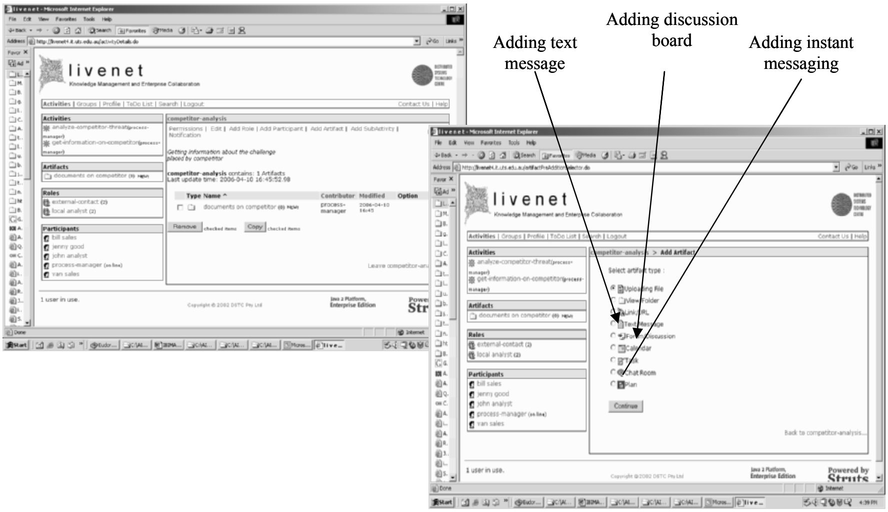


Fig. 5. Setting up lightweight exchanges

The first workspace in Figure 5 shows an initial folder containing information on the competitor. The second shows services that can be chosen to support lightweight exchange. It shows for example a chatroom that supports instant messaging as well as the ability post text messages and discussion boards.

Figure 6 illustrates progression to lightweight collaboration. The first workspace shows the workspace created for lightweight exchange using the services selected in Figure 5. The second interface shows the services that can be added to support collaboration.

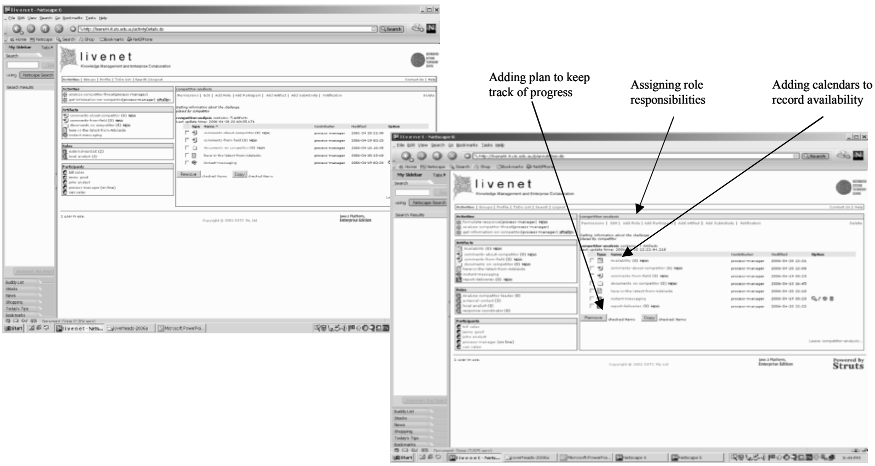


Fig. 6. Going on to lightweight collaboration

7 Summary

This paper developed a framework for choosing technical strategies to support collaborative work. It identified a number of strategies ranging from lightweight exchange to process management and the tools needed to support them. It also provided guidelines for choosing a strategy depending on organizational and cultural factors.

References

Anderson, K.M., Anderson, A., Wadhwani, w., Bartolo, L. (2003): “Metis: A lightweight, Flexible, and Web-based Workflow Services for Digital Libraries” Proceedings 3rd. ACM/IEEE conference on digital libraries, May 2003, pp. 98-109.

Cummings, J.N., Butler, B. and Kraut, R. (2002): “The Quality of OnLine Social Relationships” Communications of the ACM, Vol. 45, No. 1, July, 2002, pp. 103-111.

Grant, R.M. (1996): “Prospering in Dynamically-competitive Environments: Organizational Capability as Knowledge Integration” Organization Science, Vol. 7, No. 4, July, 1996, pp. 375-387.

Hansen, M.T., Nohria, N. and Tierney, T. (1999): “Whats your Strategy for Managing Knowledge” Harvard Business Review, March-April, 1999, pp. 106-116.

Hawryszkiewycz, I.T. (1996): “Support Services For Business Networking”, in Proceedings IFIP96, Canberra, eds. E. Altman and N. Terashima, Chapman and Hall, London, ISBN 0-412-75560-2.

Hawryszkiewycz, I.T. (1997): “A Framework for Strategic Planning for Communications Support” Proceedings of The inaugural Conference of Informatics in Multinational Enterprises, Washington, October, 1997, pp. 141-151.

Hofstede, G. (1980): Culture’s Consequences: “International Differences in Work-related values” Sage, Beverly Hills, California.

Rob Rodin Optimize Magazine http://www.optimizemag.com/issue/002/roi.htm (last accessed Jan 3, 2006).

Whittaker, S., Swanson, J., Kucan, J. and Sidner, C. (1999): “Telenotes: Managing lightweight interactions in the desktop”.

Zhang, J., Patel, V., Johnson, K., Smith, J. (2002): “Designing Human Centered Distributed Information Systems”, IEEE Intelligent Systems, September/ October, 2002, pp. 42-47.

Proposal for a System Based on the Universal Design Approach for Providing Tourism Information by Linking RFID and GIS

Akihiro Abe*, Nobuyuki Maita*, Yasunori Ooshida**, Toru Kano***

* Faculty of Software & Information Science, Iwate Prefectural Universtity, Japan
** Morioka Support Plaza for Handicapped People, Japan
*** Faculty of Social Welfare, Iwate Prefectural University, Japan

Abstract: In recent years, tourist regions have placed greater stress on the concept of Universal Design (UD), which takes into account the needs of a more diverse range of people, including senior citizens, the disabled, and foreign visitors. We have been conducting research and development on a system, based on the UD concept, for providing tourism information suited to various user characteristics. The system is designed for Hiraizumi, a tourist area which is aiming to register its historical cultural assets on the World Heritage List in 2008. This paper analyzes needs relating to Universal Design of tourist information, and describes system design and prototype development/evaluation efforts based on that analysis. Satisfactory results were obtained regarding the validity and implementation potential of our approach.

1 Introduction

In recent years, tourist regions have placed greater stress on the concept of Universal Design (UD), which takes into account the needs of a more diverse range of people, including senior citizens, the disabled, and foreign visitors [1]. As a concept which is broader than barrier-free design, UD aims to “design products, buildings and spaces so they can be used, as far as possible, by all people, regardless of age, sex or disability etc.” The Hiraizumi region of Iwate Prefecture, Japan, is aiming to register its historical cultural assets (primarily Chusonji) on the World Heritage List in 2008 [2], but some have pointed out the issue of coping with the expected growth in the number of tourists after registration. There are limits on UD and guidance improvement of facilities and signs, and, media such as mobile phones are attracting attention as a means of overcoming this problem.

Research on information support for the mobility impaired covers a wide range -- from development of basic technologies to applications in the community -- and includes support for the visually impaired within train station premises [3], and searching for movement routes using barrier-free maps [4]. In Japan, community interest has grown in recent years due to the Project for Promoting Independent Movement of the Ministry of Land [5], Infrastructure and Transport, and this project is positioned as one of the key components of the u-Japan policy [6]. Based on these previous studies, we are taking Hiraizumi as an example, and focusing on more practical, empirical research into information-related UD which takes into account the need to shift to regional information. This should be useful for verifying the regional benefits and problems of an ubiquitous network deployed in a top-down fashion as national information policy.

In this research, R&D efforts focused on a system which combines RFID (Radio Frequency IDentification)[1] with a GIS (Geographical Information System). This system allows information needed in a tourist region to be provided in accordance with UD principles to various users via mobile phone. First, Section 2 analyzes the need for UD in providing information for tourist regions. Section 3 gives the basic design of the system based on results obtained from the analysis in Section 2. Section 4 describes prototype development and evaluation.

2 Analysis of the need for UD in providing information

In order to analyze tourist needs for information, we held two workshops to consider UD in tourist regions, and administered questionnaire surveys. The 33 subjects included able-bodied persons, as well as senior citizens, disabled persons and foreign nationals. The questionnaire asked about individual attributes such as respondent age and status (disabled or not), information needs and methods of providing information.

2.1 Needed information content

On the whole, there is a significant need for guidance information at historical sites, and relatively little need for souvenir and restaurant information.

Compared to able-bodied persons, the physically disabled have special needs for toilet information. This is because wheelchair users and persons with an ostomate require information describing the location of compatible toilets, width of each toilet entrance, and the area inside the toilet.

[1] RFID is used as a term referring to wireless tags and IC tags, and for convenience we use the term ID to refer to their identification codes. The term RFID system is used to refer to a system including RFIDs together with the read/write equipment needed for to read from or write to RFIDs.

The visually impaired responded that they need almost all kinds of information. This is because they cannot rely at all on vision, and the amount of information they can obtain from the environment is quite small compared to other people. Persons whose native language is Chinese pointed out that, in addition to linguistic factors, there are differences in ease of understanding due to differences in culture. More specifically, they find it difficult to understand years indicated by Japanese era titles like “Taisho” and “Showa”. One way of addressing this problem is convert to years in the standard Western calendar, or to indicate years in a fashion suited each person’s culture.

None of the subjects of the questionnaire were children, but in workshop discussions it was pointed out that there is little content for children to enjoy. Difficult to understand pamphlets and guide signs alone are not sufficient. It would be desirable to have materials incorporating pictographic symbols, Japanese descriptions with *furigana* (Japanese syllabary) added to the *kanji* (Chinese characters), and easier-to-understand text.

2.2 Methods of providing information

Some subjects were of the opinion that fixed kiosk terminals with a touch panel are easy to use, but on the whole, people favored the idea of acquiring information by mobile phone. The following are 3 important points which were expressed regarding information acquisition:

- Historic tourism involves enjoying the atmosphere of the tourist region. Immoderately placing guide signs everywhere because there is insufficient availability of information will ruin the scenery. Subjects also expressed a strong desire to preserve the scenery of the temple. Providing information using RFID makes this possible.
- Demand is high for terminals to provide audio and text information, like the guide receivers which are frequently seen at historical sites and tourism spots. When walking through a tourist region with a guide for a group trip, able-bodied persons, wheelchair users and the visually impaired move at different speeds, and thus cannot enjoy tourism at their own pace. If guidance information on historical sites can be easily retrieved using mobile phones (the most widely available terminals), it will help to resolve this problem.
- One potentially effective approach for people with physical limitations is to provide “push-type” information using active RFID which has a built-in battery and can communicate over longer distances. Although many people use mobile phones, most senior citizens find terminal operation complicated. Disabled persons may not be able to physically approach the location where information is being provided. Another possible application is to issue a warning when a visually impaired person approaches a dangerous location.

3 System design

3.1 Design principles

Fig. 1 shows the configuration the system should have, based on the results of need analysis in Section 2. With a built-in battery, reading can be done at a comparatively long distance of up to 10m. This can be achieved by providing active RFID (which is miniaturized to preserve scenery) at key indoor/outdoor tourism spots. This enables a tourist with an RFID mobile terminal to easily search for information suited to his or her own physical and language characteristics. Information needed by tourists includes not only guide information on historical sites, but also UD information on toilets and means of movement etc.In all cases, information should be provided in relation to geographical locations, and managed with the GIS.

A server is needed to link RFID and the GIS, and we decided to call this server ITAG (Integration server of e-Tag Applications and GISs). ITAG must be able to link up with multiple tourism GISs, such as commercial and public services. While ensuring adequate protection of privacy, the system must analyze the tourism behavior of tourists from RFID access logs, and have the ability to gather access logs for use in tourist region development as well.

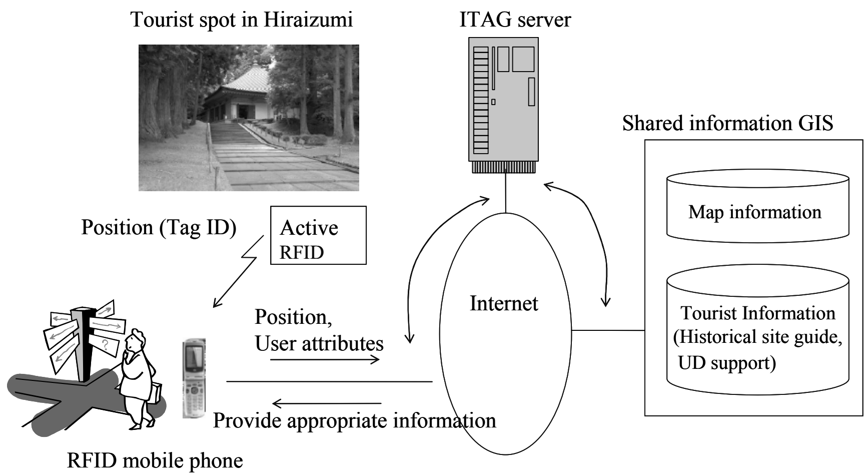


Fig. 1. Overview of system for providing tourism information

3.2 Definition of user requirements using a UD matrix

Based on the results of analyzing needs in Section 2, we define the requirements of a tourist information system based on UD principles. The requirements are organized from two sides -- information content and method of providing information -- using a UD matrix [7] for effectively extracting UD requirements and problems corresponding to the situation of each individual. User groups which bring together hypothetical users by their usage characteristics were established as fol-follows by referring to a standard user classification for a UD matrix.

1. No special consideration necessary
2. Wheelchair
3. Requires consideration of cultural differences
4. Requires consideration of ease of understanding
5. Cannot rely on vision
6. Requires consideration of vision
7. Requires consideration of hearing
8. Internal impairment
9. Consideration of advanced age

In analysis from the information content side, the requirements were organized using a matrix of the information categories required for each guide spot (historical site guide, facility management, transportation, pedestrian movement, toilets, rest areas, escape routes, and events) and the aforementioned user groups. In analysis from the standpoint of providing information, the requirements were organized using a matrix of the individual tasks involved in providing information by mobile phone (Open, Check status, Input, Select menu, Browse content, Close) and the aforementioned user groups (Fig. 2).

User Group / Individual Task	Wheelchairs	Consideration of cultural differences	Consideration of understanding	Cannot rely on vision	
Open	Easy to retrieve, with a strap etc.				
Check status		Can understand status in native language	Status is easy to understand	Can understand status with audio	
Input			Input method is easy to understand	Can perform input operation without relying on vision	

Fig. 2. UD matrix on providing information

3.3 ITAG basic architecture

Flow of server processing. Fig. 3 shows the basic architecture of ITAG. Fig. 4 shows the flow of processing up to obtaining GIS content from a tag ID acquired by the user with an RFID mobile phone. First, the tag ID and terminal-specific ID acquired by the user with the RFID reader of the terminal are sent to the IF module of the ITAG. If the authentication processing module completes authentication of the terminal-specific ID normally, a tag ID is sent to the data access module to specify the appropriate service, and if multiple selections are possible, a request for selection is sent to the user. When the user makes a selection, connection to the GIS server is secured via the data access module, and, as a result, the system searches out and stores information tied to the tag ID. The results of this processing are returned to the user via the IF module.

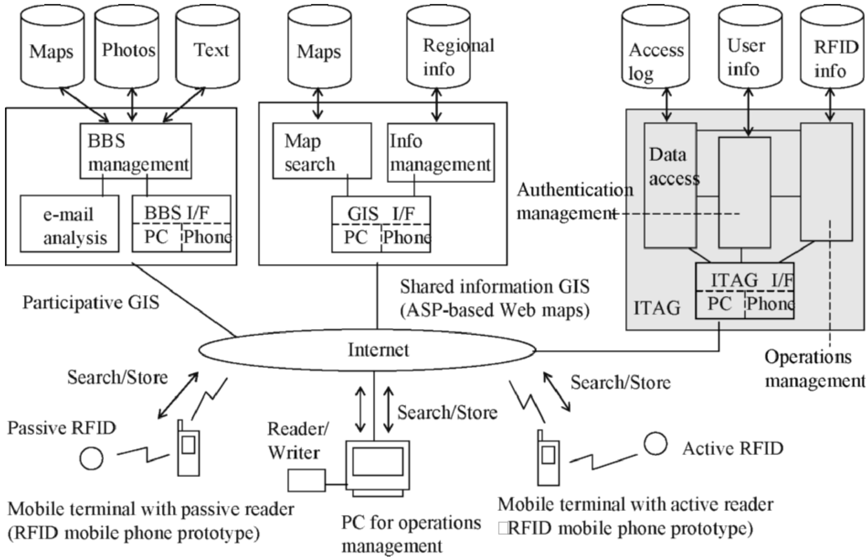


Fig. 3. System architecture

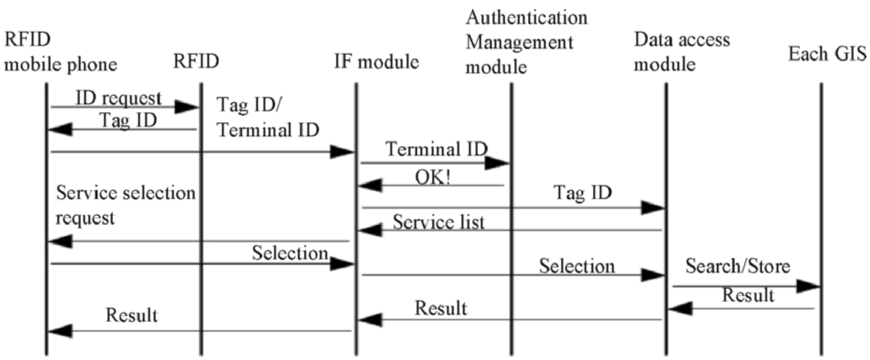


Fig. 4. Overview of processing at ITAG server

Authentication management module. For access from a mobile phone, authentication is performed using a terminal specific ID etc. If authentication processing is performed each time the ITAG server is accessed, users unfamiliar with character input using a mobile phone will be forced to go through extremely inconvenient input operation. Therefore, the system requests password input only for the first access, and then stores a terminal-specific ID for the authenticated user in the user management DB. This makes it unnecessary to perform input operation for the second and subsequent accesses. When an authenticated user is given privileges at a different level, the permitted privileges are stored together with the authenticated terminal-specific ID.

Data access module. The data access module is divided into tag analysis processing and DB processing. In tag analysis processing, processing is performed to properly select the GIS services related to the tag ID read with an RFID reader. GIS services corresponding to each tag ID are stored in the RFID management DB when the RFID is set up. If the services which can be used for each privilege are changed, the judgment of whether or not it is okay to provide information with the currently authenticated privileges is made after the service is determined. When further progress has been made in tag ID system standardization, a GIS server will be specified by accessing a global GIS server, but at the present stage, the aforementioned processing (including tag ID duplication management) must be performed on the ITAG server side.

In DB processing, logs of retrieval of management information to an RFID, or access to the GIS server, are stored in the access log DB. RFID management information is comprised of position information and information on services tied to an RFID. It is likely that the accessed tag ID, time, terminal-specific ID and GIS service are enough information for applications such as maintenance prediction for roadway structure management and analysis of tourism behavior by tourists.

Operations management module. Since a PC is used for operations management, it is necessary to have an operations management interface employing maps. When storing or updating RFID management information, it is necessary to specify the target RFID, but it is difficult for people to grasp the situation with just the numerical value for the RFID or GPS. Therefore, a management interface for the map base is essential as a means for managing the positions where RFIDs are installed.

4 Prototype development

4.1 System environment

First, we will describe the prototype development environment. As a mobile terminal environment employing RFID, we used an RFID reader equipped mobile phone test unit made by KDDI (hereafter called the "RFID mobile phone") (Table 1, Fig. 5,) [8]. RFID read control on the mobile phone side is written in Java. The ITAG was built on a work station of a university research laboratory. The OS is Solaris 9, the database is PostgreSQL and the server program was developed with PHP. As an RFID-linked GIS, we selected a citizen participation type GIS researched and developed by the authors and used for verification research in the context of various community activities [9][10], and a commercial ASP-based web map service.

Table 1. Specifications of RFID mobile phone prototype

Compatible electronic tag specifications	Individual specifications
Electronic tag frequency	315 [MHz]
Electronic tag ID bit length	64 – 128 [bit]
Electronic tag read distance	Approx. 10 [m] max
Operation count and time for electronic tag reading	Approx. 10 hours (Continuous operation time)
External dimensions of tag reader	38 [mm] × 80 [mm] × 10 [mm]
Remark	The tag ID value can be set from the application.

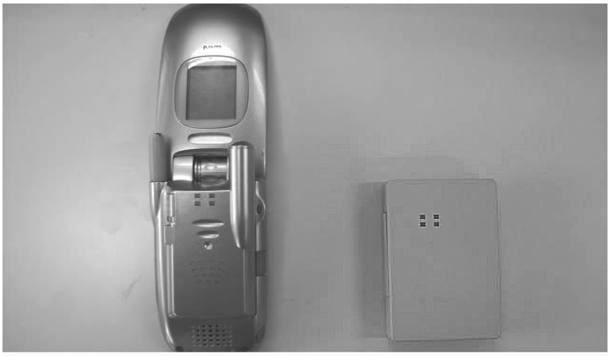


Fig. 5. RFID mobile phone prototype and active RFID

Next, we shall summarize the features of the implemented prototype. The system was designed to set user attributes relating to information acquisition using a Java application on the RFID mobile phone side (at left, Fig. 6). The attributes which can be set include: selection of language used (Japanese, English), wheelchair use, internal impairment and hearing impairment. When a user enters the transmission area of a tourism spot where an active RFID (Fig. 5, right side) is installed, the RFID mobile phone tag is received, and the mobile phone vibrates to alert the user of incoming information. When this happens, the user can select whether or not to acquire the content. If the user wishes to acquire the content, the tag ID indicating position and user attributes is passed to the GIS via the ITAG server. Content is generated on the GIS side by taking these values into account. For example, if the person is a wheelchair user, information on toilets nearest to the receiving spot is delivered to the user with data on the entrance width and a photo of the inside of the toilet (Fig. 6, right side). If the user desires, he or she can also acquire a map.

Fig. 6. Prototype screen

4.2 Field tests

We asked the Iwate Welfare GIS Promotion Study Group members, who include UD specialists and wheelchair users, to evaluate the prototype.

Installing active RFID. Key guide spots within the Chusonji Temple grounds were interspersed along an approximately 1km long approach path called "Tsukimisaka" (moon viewing hill) surrounded by rows of cedar trees . Active RFIDs were installed on both sides of the approach path, while taking into consideration the tourist traffic lines, so that a communication distance of roughly 10m could be stably secured. Experiments were also conducted at a number of exhibition guide spots within the treasure hall, where the national treasures of Chusonji are exhibited, and we confirmed that it is possible to provide comparatively fine-grained area services by adjusting the strength of transmission from active RFID.

Using the RFID mobile phone. At the receive reader on the mobile phone side, the antenna is installed vertically on the right side of the phone body, and if this part is held down by the user's hand, it will have a significant effect on reception capability. This problem can be solved by, for example, having the user to move through the area with the terminal hanging from the neck by a strap, but this is not very desirable because it places constraints on utilization by the user.

Delay of information acquisition. It would be hard to notify a mobile phone from an active RFID when a visually impaired person mistakenly enters a dangerous area, because there is too long of a delay with the current system in acquiring information from the server. It may be possible to alert users of danger at the time of terminal reception by adding information to the tag ID.

Re-browsing of acquired information. Basically, content is acquired at the actual site, but since some users may wish to later use historical site guide information accessed at the actual site for history study, or for creating memories of their trip, some expressed the opinion that they would like to re-browse information acquired at the actual site after they return home from their trip. Various applications are possible in cases where RFID and dedicated terminals have been used in museums[11]. These services enable a user to later access information quired at the actual site from a PC using a designated login ID for a fixed time after returning home. We would like to incorporate ideas like this at Hiraizumi.

5 Conclusion

In this research, we proposed a system, based on UD principles, which makes it possible to supply tourism the information needed at Hiraizumi, in a form suited to the various characteristics of users. By conducting system design and prototype development/evaluation based on analysis of user needs, we were able to obtain satisfactory results with regard to the validity and feasibility of our approach. Based on these results, we plan to conduct a pilot program in key tourist areas of Hiraizumi in 2006. In this program, we plan to have actual tourists use the system for a fixed period of time, and thereby clarify the system requirements necessary for achieving practical use. At this time, the system will not be limited only to active RFID. We also plan to study approaches like combined use with passive RFID in stamp rallies for children, and intend to conduct a comprehensive verification of effective methods of using RFID in UD approaches to providing tourist information.

References

1. Principles of universal design. http://www.design.ncsu.edu/cud/
2. The cultural heritage of Hiraizumi toward world heritage inscription. http://www.pref.iwate.jp/~hp0907/english/index.html
3. Goto K, Matsubara H, Fukasawa N, Mizukami N (2003) A personalized information system for visually disabled people using mobile terminals in railway station environment. IPSJ Journal, vol 44, no 12, pp 3256-3268 (in Japanese)
4. Eguchi I, Igi S (2005) Geographic information system for pedestrian navigation with areas and routes accessibility. IPSJ Journal, vol 46, no 12, pp 2940-2951 (in Japanese)
5. Committee for promoting project to support moving independently.http://www.jiritsu-project.jp/
6. u-Japan policy. http://www.soumu.go.jp/menu_02/ict/u-japan_en/index.html
7. Yamazaki K, Yamaoka T, Okada A, Nomura A, Yanagida K (2001) Universal design practical guideline, 1st International Conference on Planning and Design, JP009-F01-06
8. KDDI news releases. http://www.kddi.com/english/corporate/news_release/2005/0302/index.html

9. Abe A, Sasaki T (2002) A bulletin board system using geographical location information for local community activities. Proc 2002 International Conference on Information and Management Sciences, pp 31-37
10. Abe A, Sasaki T, Odashima N (2003) GLI-BBS: A groupware based on geographical location information for field workers. Proc 5th International Conference on Enterprise Information Systems, vol 4, pp 3-9
11. His S, Fait H (2005) RFID enhances visitors' museum experience at the exploratorium, Communications of ACM, vol 48, no 9, pp 60-65

Industrial Automated Fingerprint-Based Identification System

Paulius Zubavicius, Antanas Cenys, Lukas Radvilavicius

Abstract: Fingerprints recognition is established technique for police agencies. Most of the companies and public organizations are using password or token such as magnetic or smart card based identity management systems. Security of these systems depends crucially on the proper behavior of the users since passwords and tokens can be easily transferred. Biometric identification has obvious advantages if security of the system is not always in the best interest of the users. Biometrical data of the fingerprint is the best way to identify the person taking into account efficiency, availability, reliability, time-constrains, etc. Technology advances in computing and optical scanners allow create low –cost small size fingerprint-based identification systems.

The paper describes principles of the industrial Automated Fingerprints Identification Systems (AFIS). Available technologies, algorithms, usage limitations are also provided. Implemented and functioning industrial AFIS system to register and analyze company's personnel working schedule is presented.

1 Introduction

Most organizations, companies and persons keep their confidential information in digital files and databases or/and paper files locked in safes. Only identified and authorized users should be able to modify and to have access to this information. Management of ID or more specifically electronic or eID is becoming a very significant part of the most of information systems. Shift from paper files to electronic files and databases resulted in the password based ID management and security systems as a standard. These systems are based on the identification by "what do you know" principle. Introduction of magnetic and smart cards as well as other type of electronic tokens added a possibility to identify person electronically by "what do you have" principle corresponding to traditional lock and key system.

These systems have, however, limitations since they could not guarantee the identification of the person. Both passwords and tokens can be stolen or voluntary transferred to another person. The only way to ensure identification of the person is to use biometric data unique for this person. This is identification by "who you are" principle. Biometrics-based authentication is emerging as an alternative method. It can overcome some of the limitations of the traditional automatic personal identification technologies because of dealing with physiological and/or behavioral characteristics, such as a fingerprint, signature, palmprint, iris, hand, voice or face [1]. This data can be used to authenticate a person's claim to a certain identity or establish a person's identity from a large database. The need for secure transaction processing using biometrics technology will be only growing in the future based on the rapid evolution of electronic commerce and other applications with increased emphasis on security [2].

The most developed and widely used biometric ID management systems are based on fingerprints recognition. Automated fingerprints identification systems (AFIS) can search for particular pattern in the large databases with more that 100 000 fingerprints as "one with many" identification. However, application of AFIS for "one-to-one" verification is faster and more common [3].

Until very recently fingerprint technologies were widely used in military, national security, police and similar institutions. Only now such technologies along with other biometric techniques are starting to be used it civil and commercial sectors. The main reasons for slow adoption in commercial systems were high costs and limited availability. Technology advances in technology and computing have changed the situation radically and nowadays fingerprint technology is becoming affordable even for small industrial ID management systems [3].

In the paper we present an implemented and functioning application of AFIS for registration and analysis of personnel's working schedule. The paper describes principles of the industrial Automated Fingerprints Identification Systems for this specific application, available technologies, algorithms, and usage limitations.

2 Biometric Identification

There are two different ways to resolve a person's identity: verification and identification. Verification is based on confirming or denying a person's claimed identity and answers question "Am I whom I claim I am?". In the case of identification one has to establish a person's identity answering the question "Who am I?". A biometric system is essentially a pattern recognition system allowing compare unique physiological or behavioral characteristic possessed by the user with prerecorded data. As such it can be used in both above approaches to identification [2]. The method how an individual is identified is very important, however, while designing a real world system.

Identification is "one-to-many" process of determining a person's identity by performing matches against multiple biometric templates. There are two types of identification systems: positive identification and negative identification. Positive identification systems are designed to find a match for a user's biometric information in a database of biometric information and answers the "Who am I?" question even though the response is not necessarily a name. It could be an employee's ID number, nickname or any other unique identifier. A prison release program where users do not enter an ID number or use a card would be an example of a typical positive identification system. Users then simply look at an iris capture device and are identified from an inmate database. The negative identification systems also compare one biometric template against many in the database. But, in fact, they are designed to ensure that a person is not present in a database in order to prevent people from enrolling twice in a system. It is often used in large-scale public benefits programs in which users enroll multiple times to gain benefits under different names [2].

The oldest method among all the biometric techniques is fingerprint-based identification which has been successfully used in numerous applications. Everyone has unique, immutable fingerprints which are made of a series of ridges and furrows on the surface. The pattern of ridges and furrows as well as the minutiae points determines the uniqueness of a fingerprint [4]. Minutiae points (Fig. 1) are local ridge characteristics that occur at either a ridge bifurcation or a ridge ending [5].

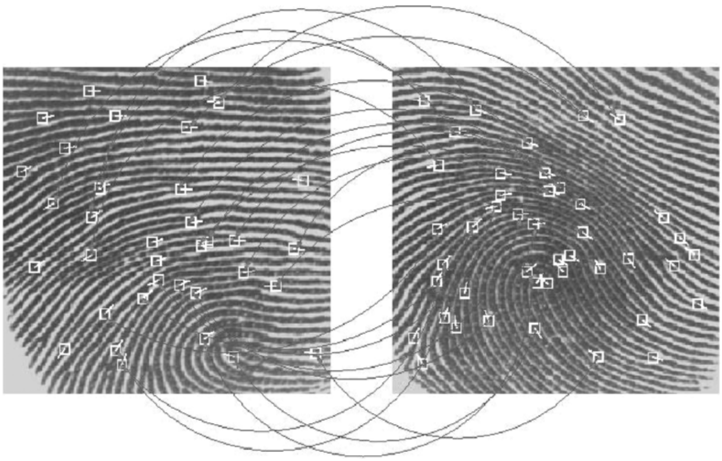


Fig. 1. Minutiae points

Fingerprint technology acquires the fingerprint pattern, but doesn't store the full image. Some particular data about the fingerprint is stored in a much smaller template or minutiae point data, requiring from 250-1000 bytes [6]. In the our particular system approximately 380 bytes of data are stored. The full fingerprint cannot be reconstructed from the fingerprint template and it is not stored after the data is extracted [4].

World-wide used fingerprint technology serves hundreds of thousands of people daily to access networks and PCs, enter restricted areas, and to authorize transactions. The technology is used broadly in a range of applications, primarily PC/Network Access, Physical Security/Time and Attendance, and Civil ID. Most deployments are "one-to-one" verification systems, but there are also a number of "one-to-few" deployments in which individuals are matched against modest databases, typically of 10-100 users. Large-scale "one-to-many" applications in which a user is identified from a large fingerprint database searches on databases of up to millions of fingerprints. Depending on computational power of Automated

Fingerprints Identification Systems (AFIS) these searches can be performed within only a few minutes [2].

3 Fingerprint Identification Based Employee Management System

Most of the companies and public organizations are using password or token such as magnetic or smart card based identity management systems. Security of these systems depends on the proper behavior of the users since passwords and tokens can be easily transferred. Such behavior can be expected only if it is in the own interest of the users to keep usernames, passwords, password cards and password generators safe. For example in internet banking systems users have access data to login and manage bank accounts. In case of loosing it users can have serious troubles or even loss of money.

In case of access control and employees management systems users can have serious incentives to give access data or tools to third party. Automated personnel management system described bellow is recording in/out time of employee. Since employees are obviously interested to have a right time records there is a place for cheating. Most of the automated access control systems use magnetic cards. Serious disadvantage of these systems is the possibility for user or employee to provide his card for another employee to make faked records. Moreover the same identification card can be used by several employees or several cards can be used by same person. There is no simple electronic system to exclude such possibility. This problem can be solved involving human control, however, this significantly increases costs and still one can have problems with human factor. Other tools used in automated identification process such as passwords, electronic tokens, smart cards, RFID tags are also transferable and sometimes can be even duplicated. Biometric and particularly fingerprint identification has obvious advantages.

Features of the different identification methods are summarized bellow in the Table 1:

Table 1. Features of the different identification methods.

Features	Password	Card	Human control	Finger print reader
Cheating possibility	+	+	+	-
Time consuming	+	-	-	-
Salary costs	-	-	+	-
Extra requirements	+	+	-	-
Falsification	-	+	-	-
Duplicate	+	+	-	-

The advantages of the fingerprint based identifications become evident while such identification is applied for work time accounting and control system at organization with more than 100 employees with variable schedule. Control of the work time of employees on a constant schedule is not difficult but often there are people working on a variable schedule including late hours, weekends, which is not easy to control. The magnetic card system is not effective since it does not guarantee person's identification.

We have developed and installed automated fingerprint-based personnel control system. The system works like timecard system and provides access control, work time accounting, fill time-board and etc. The system includes three components (see Fig. 2):

- Enrollment place
- Administrative place
- Data Base Server

The system is developed using Borland Delphi 7 IDE. All components run on separate PC's connected to TCP/IP network. The schematic layout of the components is shown in Fig. 2:

Fig. 2. The structure of the system.

3.1 Data Base Server

Fingerprints meta-data, employee's names, ID and related information is stored in the data base. The data base server is running "Gentoo" Linux OS because it's powerful and fast operating system for servers. PostgreSQL DBMS running in this server is an open source product. Data base is created in already existing DBMS and connected to a particular already running system. DBMS and data base have no identification function.

3.2 Administrative place

Administrative place is used to control the system and generate reports. It can be accessed and run remotely.
The main functions of this component are:

- Possibility to enter and edit employee's data in the database.

- Creation of both variable and constant working schedule.
- Reporting on employee lags.
- Reporting on working time, free days, vacations and so.

Other functions include:

- Possibility to edit real events if an employee made a mistake. It is available only for authorized employees.
- Possibility to know who is working at the moment and who is not.
- Each change or editing in scheduling is authorized.

3.3 Enrollment place

This component includes the hardware scanning fingerprints. The scanned fingerprint image is processed by the fingerprints hardware driver with the graphic algorithm. That we could get minutiae points' fingerprints identification and application takes these data from the hardware driver and places it in the data base according to the particular employee's ID (Fig. 3). Subsequently these data are applied for identification.

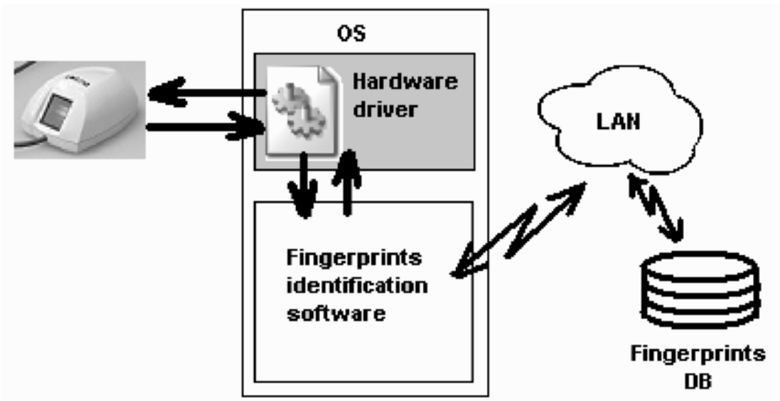


Fig. 3. The fingerprints identification.

All fingerprints data are read from the data base to dynamic array. The whole identification process including reading a fingerprint and comparing with the data in the dynamic array is shorter than one second. After the identification process in the data base is created record of employee's entry and escape. UML diagram of the working principle is presented in Fig. 4:

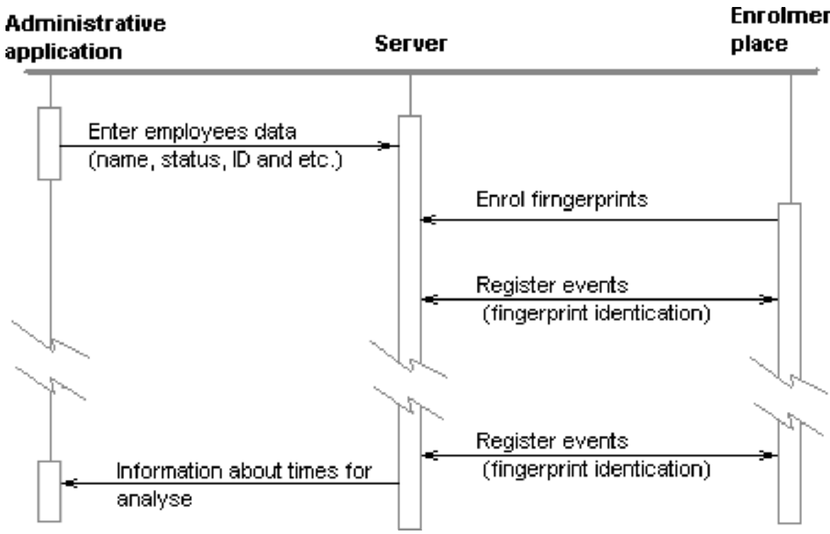


Fig. 4. Fingerprints system's UML diagram.

The employees are identificated each time they enter or leave a building. All the check-in's are recorded and compared to the pre-set schedule.

The system is installed and functions without any significant failures. Some times fingerprint reader fail, because fingerprint was damp or dusty. Employees with smaller fingers can have problem while reading fingerprint since reader does not get full image. These problems were eliminated adjusting fingerprint reader's quality and security parameters.

4 Conclusions

We have developed and installed functioning fingerprint-based system to automate the employees control process. It allows easily and conveniently monitor a real working time schedule and helps the executives to administrate the organization. The system is an example of successful application of biometric technology and integration of AFIS into the already existing and running control system.

References

1. **Biometric Consortium** (2005-12-28) Introduction to biometrics, http://www.biometrics.org/html/introduction.html;
2. **ROSISTEM** (2005-12-28) Biometrics, http://www.barcode.ro/tutorials/biometrics/;

3. **Sgt. David R. Ashbaugh**(2006-01-04). Automated Fingerprint Identification System (A.F.I.S.), http://www.ridgesandfurrows.homestead.com/afispage.html;
4. **Salil Prabhakar, Anil Jain** (2005-12-28) Fingerprint Identification, http://biometrics.cse.msu.edu/fingerprint.html;
5. **Dr. O.P. Jasuja** (2006-01-03) Poroscopy: a method of personal identification revisited, http://www.fingerprints.tk/;
6. **Sgt. David R. Ashbaugh** (2006-01-04). Friction Ridge Identification, http://www.ridgesandfurrows.homestead.com/identification.html;
7. **Jean-François Mainguet** (2005-12-28). Biometrics, http://perso.wanadoo.fr/fingerchip/biometrics/biometrics.htm;
8. **Security Database B.V.** (2006-01-04). Large Scale Automatic Biometric Identification System, http://www.securitydatabase.net/megamatcher.html;

Database Architectures: Current Trends and their Relationships to Requirements of Practice

Jaroslav Pokorný

Charles University, Faculty of Mathematics and Physics, Praha, Czech Republic, pokorny@ksi.ms.mff.cuni.cz

1 Introduction

Due to changes in the world of data, new demands on databases appear and consequently questions, where the databases field is and where it should be going. Abiteboul et al. [1] emphasize two main driving forces in database area: Internet and particular sciences, as the physics, biology, medicine, and engineering. These sciences produce large and complex data sets that require more advanced database support than current commercial systems provide. For example, BaBar database, containing nuclear data and considered as the biggest in the world, had more than 895 terabytes data stored in 847149 files on November 5, 2004. The system CORIE (Columbia River Estuary) produces in its simulations 5 gigabytes of forecast data each day [5]. Data volume doubles approximately every year and it is measured even in petabytes [8].

Another trend, existing since 60s, concerns industries responding to ever increasing environmental demands from customers, authorities, and governmental organizations. Also requirements on safety are increasing. In both cases the development tends towards an emergence of new monitoring systems. New functions are integrated into usual business information systems, governmental systems, and special systems, for example, digital "smart" home management systems.

Problems with processing data considered are well documented on example of environmental data. When users want to search and use environmental information, the following problems occur [17]:

1. Data does not exist or is insufficient; sometimes this may require synthesis or reproduction of data.
2. Data is not referenced by data suppliers and therefore hard to locate, or data is referenced under specific classification criteria that are domain-specific.

3. Data is hard to access; it is either private, or of a too high cost, or requiring costly pre-processing (for example, data must be re-entered manually from paper documentation) or format translation.
4. Accessed data sets are hard to use because they are inconsistent or non-compatible.
5. The quality of retrieved data is hard to assess; it is often hard to compare data produced using different scientific models because of a lack of documentation about the underlying computational processes.

Database community is focused on software architectures – *database management systems* (DBMS). The DBMS development is always driven by new applications, technological trends, and new synergy among associated fields, as well as by innovation in the field alone. Problems (1) - (5) are natural part of today's database research and development. New database technologies deployed in practice can overcome these problems.

DBMS development is influenced by a number of technological aspects. For a moment, we focus on scientific data. Sensor networks producing a part of this data consist of a large amount of cheap equipment, from which each is a data source measuring a quantitative quantity, for example, objects location or surrounding temperature. Processing such data usually distinguishes from that used in enterprise databases. Data occur in high-speed streams and queries over these streams need to be processed in online manner, which enables a response in real time. This data is moreover uncertain or imprecise, any measurements contain usual errors. Another aspect of this data includes unclear formulation of queries based on traditional techniques, as they occur, for example, in traditional databases. Often we are not able to formulate query in SQL even though we are convinced that it would be possible. In such situations apparently semantics are missing. To describe data semantics on the metadata level (formally, if possible) is more than desirable.

The purpose of the paper is to present main directions in development of database architectures reflecting an emergence of new data types and new kinds of their processing. In Section 2 we briefly describe a layered DBMS architecture as it was designed by Härder and Reuter [9] in the 80s. We consider it as referential, since any other development of databases meant to be successful in its realization. In Sections 3-7 we briefly discuss technologies and requirements of practice, which essentially influence today's database architectures; sensor data and sensor networks, processing data streams, considering uncertain and imprecise data, data mining and OLAP as well as wireless data broadcasting and mobile computing. Section 8 is already devoted to new database architectures. In Section 9 we generalize our considerations with the help of the notion of dataspace. Conclusions briefly summarize the ideas presented.

2 Multi-layered DBMS architecture

Everybody, who uses a relational database, is aware of the fact that tables occurring on the top of a database system are in some sense virtual. More specifically, they provide a logical data structure suitable for user-oriented processing data in a database. Tables are only one, most visible layer of a database system. Härder and Reuter [9] proposed a mapping model consisting from five abstraction layers. Table 1 adopted from [10] shows these five layers in detail. We can observe objects to be dealt with at each level and particular functions implementing mappings between two consecutive layers. For example, nonprocedural access in L5 layer provides tables and statements for their manipulation formulated usually in SQL. Layer L2 ensures dividing of linear address space of external memory into different types of pages. Among objects in layer L3 we can find data structures supporting indexing, for example, B-trees for strings and numbers or R-trees for spatial data. Going upwards, the objects and associated operations become more complex, also additional integrity constraints can appear.

Table 1. Description of the five-layered DBMS mapping hierarchy

	Level of abstraction	Objects	Auxiliary mapping data
L5	Nonprocedural access	Tables, views, rows	Logical schema description
L4	Record-oriented, navigational access	Records, sets, hierarchies, networks	Logical and physical schema description
L3	Record and access path management	Physical records, access paths	Free space tables, DB-key translation tables
L2	Propagation control	Segments, pages	Buffers, page tables
L1	File management	File, blocks	Directories

The concept a multi-layered architecture considers its ideal implementation with a machine, which has *k* layers. Although the number five is considered as a good compromise in the architecture, problems appear with its performance in practice. Simplification of layers complexness on one side increases run-time overhead on the other side. In consequence, various ways how to optimize DBMS performance are developed and the number of layers is for some system functions reduced.

Development of the L5 layer during last 10 years resulted in a specification of so-called *object-relational* (OR) *data model*. Its part is standardized in the standards SQL:1999 and SQL:2003. In the OR data model tables can have structured components of their rows, columns can be even of a user-defined type. Spatial data, time series or texts belong to this category. For some data types, for example VITA (video, image, text, audio), there is a standardized set of predicates and functions for manipulation of their instances. Such "extendible" approach resulted into so-called *universal DBMS* in the late 90s. The kernel of these database engines was extended by loosely-coupled additional modules (components) for each new data type. Vendors of leading DBMSs call these components *extenders*, *data blades*, and *cartridges*, respectively. Remind, that spatial and text components belong to the most successful results in this approach. Due to more complex data structures than rows of traditional relational tables, OR DBMS provide a chance for use in the field of storage and processing scientific data, where the structure ARRAY belongs among key data structures.

The possibility of user-defined types brought in implementation of DBMS architectures a lot of serious problems, particularly in case of conceptually wholly different data types, as e.g. VITA. For VITA types it is possible to use together maximally the layer L1, the other have to be implemented for each type separately. An open problem remains how to integrate these types into a common framework. Implementation of new access paths, like special types of indices, results usually in modification of the DBMS kernel, for example, SQL compiler, query optimizer, etc. Such changes are very expansive, time-consuming and tending to errors.

Each vendor uses to open the architecture of host system to some extent a different approach. Oracle cartridges are restricted to integration of secondary indices. In IBM DB2 extenders there is a framework for indexing new data types limited only to B-trees. It means that such indexing can bring an improvement of evaluation only for some types of queries. In other words, new functionality is supported, but only for a restricted class of user requirements.

It seems that the benefit of the universal database software is apparent mainly in case of requirements, which can be decomposed into relatively independent parts evaluated separately in the DBMS kernel and in the module, which implements a specific data type. Processing frameworks are either too complex or not flexible enough to cope with a wide range of user requirements on domain-specific access methods. In fact, a seamless integration can be hardly achieved with these attempts. Today's implementations of layered DBMS architectures are not sufficient for new requirements and fail in the case of universal DBMSs.

3 Sensor data and sensor networks

Among new ITC technologies the inexpensive micro sensor technology is number one in context of new requirements on databases. Sensors enable most objects to report messages about their attributes, as, for example, temperature, pressure, state or location, in real time. This information will support applications, whose main purpose is to monitor such attributes [3].

Sensor networks produce important data sources and create new requirements on data management. In fact, they become a new kind of a database engine, whose optimal use requires operations to be pushed as close to the data as possible. Generally, sensors and/or users can be even mobile.

A sensor information processing is establishing the most interesting database problems. Large data collections generated by sensors will be distributed through the world, and their data will come and go dynamically. Sensors can produce continual, possibly unlimited, data streams.

At first sight, sensor networks are similar to distributed databases extended by features related to real time. There is an important difference, that a degree of evaluation of data created in a sensor network is higher, than it is considered in distributed DBMS. This breaks the traditional information integration paradigm, since there is no practical way to extract and load data into a common database to each such occurrence. Strategies for query processing and optimization have to be redefined as well.

4 Data streams processing

Management of data coming from sensors based exclusively on traditional "store-and-query" model as it used in today's DBMSs usually can not effectively deal with volume and velocity of streaming data, whose values might exist a moment [3]:

- sensor nodes produce and deliver data continuously without receiving requests for that data,
- queries over collected data can be less frequent than data insertions,
- produced data has often to be processed in real-time because it can represent events, that need a rapid answer,
- queries run continuously because data streams never terminate, so, they can see system conditions changes during their execution,
- an entire stream can not be stored on disk, and
- if the data to be processed is not available, then operators must process data only when nodes make it available.

In consequence, *Data Stream Processing Systems* (DSPS) have been developed; see, for example, [6]. A *streams processor engine* for data processing is then an example of a new database architecture, that enables the execution of queries, computations, and actions on streaming data in real-time. Such engine should accept stream oriented, continuous queries formulated in the SQL. In DSPS, a data processing is done mostly in main memory, disk operations read and write are optional and can be in many cases handled asynchronously. For example, in StreamBase DSPS, developed by Stonebraker [16] in 2005, it is possible to analyze 140,000 messages/s, while a leading relational DBMS could handle only 900 messages/s.

5 Approaching uncertain and imprecise data

In addition to data management issues of data streams, many other problems arise. Any measurement is usually a subject of errors. For example, location data for moving objects involves uncertainty concerning the current objects position. Individual sensors are not reliable; therefore wireless communication is also not reliable

Various approaches are used to provide more accurate estimation about object values, like fuzzy sets or Dempster-Shafer theory and other techniques of artificial intelligence [12].

Traditional DBMS were applied to processing enterprise data, which is typically represented by numbers and character strings. Data items are exact quantities - address, quantity on hand, delivery date, balance, and employment status. In consequence, current DBMS have no tools for processing approximate data and imprecise queries. Also sequences and images require approximate processing based on similarities, metrics, etc.

To increase data quality, new data models appear that preserve the origin of data and history of its processing, so-called *lineage* of objects and processes [4]. Data producers should include such data lineage (and authenticity) of information into metadata attached to basic (measured) data to ensure its best utilization. Other application of data lineage can be a system of personal data, where we want to preserve various document versions, related e-mails, etc.

6 Data mining and OLAP

Measured data needs to be analyzed in most cases, to obtain information necessary for decision making. A typical example is an environmental management. Comparing to simple forms of regularities or irregularities determined with statistical methods (OLAP), data mining methods can explore more complex hypotheses.

Historically, data mining focused on effective ways of discovering models of existing data sets. These models have to reveal some useful aspects of data, while obscuring details not useful for the intended application. Researchers developed algorithms, which perform operations as classification, clustering, discovering associate rules, and summarization. These techniques become new parts of products of main DBMS vendors and most of them are applicable in the field of scientific data. Unfortunately, many algorithms are super-linear (for example, for processing n points they are of complexity $O(n^2)$ or $O(n^3)$), which in case of collections mentioned in Section 1 can be unacceptable. A solution is, for example, approximate algorithms and use of parallelism.

Current interest in combination of data mining technologies and DBMS tends to discover new approaches how to store data sets which have to be mined to optimize data mining. The collection size is not issue of only scientific data. According to the Greg law, the volume of enterprise data doubles every 9 months. Data warehouses ranging in terabytes volumes are no exclusion now. Forrester [18] even estimates that most large enterprises have petabytes for data in all data repositories across the organization – and it is likely to grow to exabytes in the coming years.

Current research directions concerning mining stream data include:

- multi-dimensional OLAP for discovering unusual patterns;
- mining clusters and outliers for discovering unusual patterns; and

- single-pass classification methods for stream data mining.

7 Wireless broadcasting and mobile computing

Data broadcasting is an attractive alternative to approach "on demand", because it can broadcast data simultaneously to a large number of clients at a fixed cost. It is suitable for services based on an objects location, which exhibit strong temporal and spatial locality in that near clients in a certain time period, tend to seek the same kind of information [19].

Data to be broadcasted includes also sensor data. Sensors deployed in an environment can broadcast their data periodically or when an interested event happens. Unlike to traditional computing, client devices cannot make requests to sensors for the data. Instead, client devices just listen to the broadcast channels passively. Thus, the sensors have the initiative in communication. Sensors may broadcast data periodically, if they are measuring a continuous phenomenon producing data, or may broadcast data only when a particular event occurs, if they are detecting whether an RFID tag has come into range.

Higher-level sensors in a sensor network can pre-process low-level sensor data and then broadcast this derived information to client devices. Such processing can require modified database techniques to be successful.

It seems that location in spatio-temporal space becomes an important property of data and introduces a new dimension for data access methods. Traditional data access methods are not suitable in this case. The goal of current research is to redefine some well-known techniques, for example, processing space queries, into mobile environment with a special emphasise on data broadcasting.

8 Towards new database architectures

Database technology seems to be fundamental for deployment of technologies introduced in Sections 3-7 in context of new applications. Main benefits of the database approach should include: flexibility without complexity and ease of use. Database approach brings the opportunity to link all data together on a user level and simplify its analysis, for example with the help of technologies like data mining.

A common view on mentioned issues concerns the DBMS architecture. Today's DBMS provide practically universal architecture applicable to many various types of tasks, i.e. by words of Stonebraker and Çetinteme [15], "*one size fits all*". In new DBMS architectures rather separated database servers "*made to measure*" are expected, in accordance with requirements of particular applications. Besides traditional fields, as OLAP, data warehouses, and text retrieval, other candidates for separate engines are:

- data streams processing,

- sensor networks,
- scientific databases, and
- native XML databases.

We have tried to highlight some characteristics of the first three technologies. Considering native XML databases, solutions with a separate engine are popular today. In [10] Härder presents XTC architecture (XML Transaction Controller), which proves that native XML DBMS can be implemented in frame of five-layer architecture. There is also a possibility of so called hybrid engine. To integrate relational and XML data, IBM develops a new hybrid DB2, code-named Viper, enabling to work with a native XML store that is placed side by side with a relational data repository. On the top of both data stores sits one hybrid database engine. Similar solutions are used by many vendors, who combine a data warehouse and a usual transactional DBMS united by common parser. Such architecture can be inspiring for implementation of non-traditional data types.

Another approach evolves the idea of DBMS extendibility. Acker et al. [2] developed an Access Manager specification, a programming interface to several layers of a DBMS kernel. This enables the programmer to add new data structures to the DBMS with a minimum of effort.

There is also a third approach to achieve a flexibility of processing data in a database way: to produce a storage engine that is more configurable so that it can be tuned to the requirements of individual applications [14]. In principle, a solution must possess two features to cover the wide spectrum of today's application needs: modularity and configurability.

Modular DBMS must allow to the developer to use or exclude some subsystems depending on whether application needs them. DBMS must be configurable with respect to its operational environment: specific hardware, operation system and applications, which are using it.

9 From databases to dataspaces

All efforts on new DBMS architectures indicate, that current requirements on data management can not be dealt with storing the data into a database of one (in the best case relational) DBMS. There is a bias to place data rather into loosely coupled data sources, some of them are managed by a relational DBMS, but the other not at all. Data sources can be considered as members of a *dataspace*. Dataspaces are not another data integration approach. Data in a dataspace rather co-exists; semantic integration is not a necessary condition for operating parts of a system. Figure 1 adopted from [7] shows a categorization of existing data management solutions in two dimensions. "Administrative Proximity" indicates how close the various data sources are in terms of administrative control. "Semantic Integration" is a measure of how closely the schemas of the various data sources have been matched.

The notion of dataspace is a new abstraction described in [7]. A development of associated software – DataSpace Support Platform (DSSP) – is mentioned today as a main item of program in the field of data management, or as we often say, data engineering.

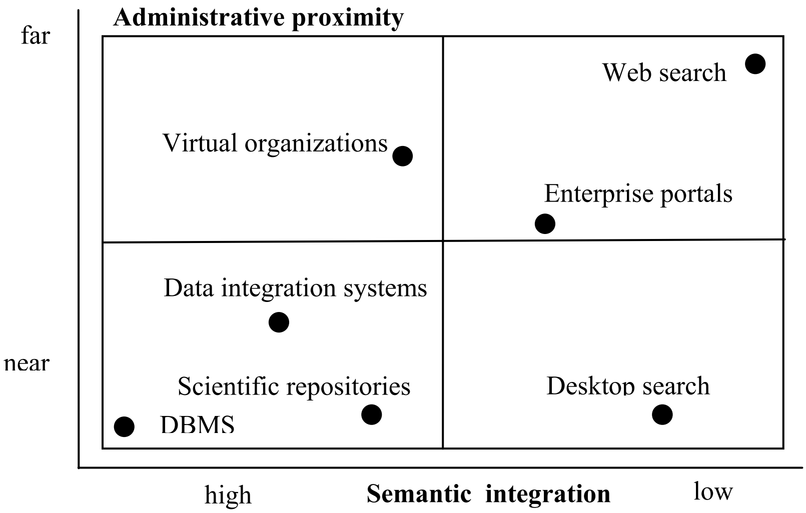


Fig. 1. A space of data management solutions

DSSP does not assume a total control on data in dataspace. It allows managing data by particular system under which belongs. It provides, however, new services over aggregations of these systems. DSSP includes a catalogue containing descriptions of *participants* (sources) in dataspace and *associations* among them. Participants are sources as, for example, relational databases, XML data repositories, text databases, web services, sensors producing data, etc. Dataspace assumes a possibility to model arbitrary association among participants, in extreme case a data exchange schema between sources. Often the relationship is simple, for example, one source is only a version of another. Integration in a dataspace is permanently evolved, on the base of demands; nevertheless data are still accessible, for example, in the simplest form via key words.

Querying includes both data and metadata, such as "Where I find something about ISD conference?", "Which sources have the attribute length?" Approximate answers are assumed, or "best effort" results in case of unavailability of sources. DSSP should also enable the user to come, if necessary, to the query language of the data source (if possible). It would be interesting to compare this proposal with semantic web, which is based on ontologies and URLs. In a dataspace we want achieve looser integration focusing rather into the depth of not necessarily web sources.

10 Conclusions

Certainly there are other criteria how to classify and evaluate new solutions in database architectures, than those discussed in the paper. We can mention, for example, the Service Oriented Database Architecture (SODA) developed by Microsoft and realized in SQL Server DBMS 2005 [11], technological solutions such as Asymmetric Massively Parallel Processing (AMPP) architecture or grid architectures.

We have seen that the trend of stand-alone database engines is rather against the previous effort to integrate all into one universal system. The concept of dataspace supports this direction. On the other hand, integration is considered rather on a global level, over data sources, based on metadata management. This feature is observable both in dataspaces and a semantic web. Possibly, the former considers integration more loosely, on demand, in comparing to the latter.

These observations are included also in challenges, which have been formulated by P. Selinger in [13]:

- re-examine DBMS architecture and invent ways to scale more and better, without sacrificing user-visible availability or performance,
- learn what managing content is all about, what is needed and create new models,
- treat metadata as and first class research.

In any case, databases architectures play a significant role in any data processing today. Everything indicates that the development of new database technologies has and will have consequences that influence future data-oriented systems.

11 Acknowledgement

This research was partially supported GACR grant 201/06/0756.

References

1. Abiteboul, S., et. al.: Lowell Database Research self-assessment. In: Communications of ACM, May 2005/Vol. 48, No. 5, 2005, pp. 111-118.
2. Acker, R., Pieringer, R., Bayer, R.: Towards Truly Extendible Database Systems. In: Proc. DEXA 2005 Conf., LNCS 3588, Springer-Verlag, 2005, pp. 596-605.
3. Amato, G., Caruso, A., Chessa, S., Masi V., Urpi, A.: State of the art and future directions in wireless sensor network's data management. 2004-TR-16, ISTI, 2004.
4. Bose, R., Frew, J.: Lineage Retrieval for Scientific Data Processing: A Survey. ACM Computing Surveys, Vol. 37, No. 1, 2005, pp. 1-28.

5. Bright, L., Maier, D.: Deriving and Managing Data Products in an Environmental Observation and Forecasting System. In: Proc. Conference on Innovative Data Systems Research (CIDR), January 2005, pp. 162-173.
6. Cherniack, M., Balakrishnan, B., Balazinska, M., Carney, D., Çetintemel, U., Xing, Y., Zdonik, S.: Scalable Distributed Stream Processing. In: Proc. of Conference on Innovative Data Systems Research (CIDR), January 2003.
7. Franklin, M., Halevy, A., Maier, D.: From Databases to Dataspaces: A New Abstraction for Information Management. ACM SIGMOD Rec., Dec. 2005.
8. Gray, J., Liu, D.T., Nieto-Santisteban, M., Szalay, A.S., DeWitt, D., Heber, G.: Scientific, Data Management in the Coming Decade. Microsoft Research, MSR-TR-2005-10, 2005.
9. Härder, T., Reuter, A.: Concepts for Implementing and Centralized Database Management System. In: Proc. Int. Computing Symposium on Application Systems Development, March 1983, Nürnberg, B.G. Teubner-Verlag, 1983, pp. 28-104.
10. Härder, T.: DBMS Architecture - Still an Open Problem. In: Proc. BTW, Karlsruhe, March 2005, pp. 2-28.
11. Kiely, D.: How SQL Server 2005 Enables Service-Oriented Database Architectures, 2006. Available: http://www.microsoft.com/technet/prodtechnol/sql/2005/sqlsoda.mspx#EHE
12. Ramamritham, K., Son, S. H., Dipippo, L.C.: Real-Time Databases and Data Services. Real-Time Systems, Kluwer AP, 28, 2004, pp. 179-215.
13. Selinger, P.: Five Data Challenges for Next Decade. Key note of ICDE conference, April 2005, Tokyo, Japan.
14. Seltzer, M. I.: Beyond Relational Databases. Databases, Vol. 3, No. 3, 2005, pp. 50-58.
15. Stonebraker, M., Çetintemel, U.: "One Size Fits All" An Idea Whose Time Has Come and Gone. In: Proc. Conf. ICDE, April 2005, Tokyo, pp. 2-11.
16. StreamBase Systems, Inc.: StreamBase™ 2.0, 2005. Available: http://www.streambase.com/index.html
17. Tomasic, A., Simon, E.: Improving Access to Environmental Data using Context Information. ACM SIGMOD Rec., Vol. 26, Issue 1, 1997, pp. 11-15.
18. Yuhanna, N., Gilpin, M.: Information Fabric: Enterprise Data Virtualization. Forrester, January 9, 2006.
19. Zheng, B., Lee, D.L.: Information Dissemination via Wireless Broadcast. In: Communications ACM, May 2005/Vol. 48, No. 5, pp. 105-110.

The IT Culture as an Obstacle to the Adoption of an ERP: Case of a High-Tech SME

Meissonier, R., Ph.D., Houzé, E., Ph.D, Belbaly, N., Ph.D.

GSCM, Montpellier Business School, France, 2300, Avenue des Moulins, 34185 Montpellier Cedex 4, CREGO, Montpellier University, France
Place Eugène Bataillon, Bâtiment 19, 34095 Montpellier Cedex 5, GSCM, Montpellier Business School, France
2300, Avenue des Moulins, 34185 Montpellier Cedex

Abstract: The firm's IT experience is considered among the characteristics that influence the adoption of an ERP. Few authors assume that the firms IT experience act as a main risk factor for ERP implementation projects. In this paper, a literature review based on theories related to resistance and conflicts factors in ERP implementation is presented. Then we describe the outcomes that may be deducted from a high-tech SME which tried to adopt an ERP. We find in this case that IT expertise and culture turned out to be inhibitors of ERP adoption. In conclusion, we present a dual perspective on the firm's capability to implement an ERP.
Key words: ERP, resistance, conflicts, culture, IT expertise

1 Introduction

For more than 10 years, a lot of companies have implemented ERP's to update their information system and to benefit from an integrated functional infrastructure. In Information Systems research, these integrated applications have been the subject of many research projects mainly focused on large firms. In the present time, ERP software editors' are targeting more and more SMEs with adapted solutions in order to modernize their applications or even to allow them to be interfaced with the ones of their customers or suppliers. Among the characteristics considered as success factors in the ERP adoption, the experience of IT firm's projects hold the attention of many authors (Lapointe & Rivard, 2005; Akkermans & Van Helden 2002; Markus *et al.*, 2000; Grover *et al.*, 1995; Wilcocks & Griffths, 1994; Barki *et al*., 1993). These studies put forward resistance to change as a main risk factor, where resistance to change is composed by the professional habits and culture of the employees.

The case study is dealing with a high-tech SME specialised in broadcast technology for media corporations (TV and radio channels). When the firm tried to adopt an ERP, its expertise in IT project management turned out to be an inhibitor because of the professional culture associated to this expertise.

The paper is organized as follow: First, a literature review based on ERP implementation theories related to resistance and conflicts factors is presented. Second, we analyse a high-tech SME case where the lessons that can be drawn are based on eight semi-directive interviews. Finally, the conclusion is focusing on the dual perspective of the firm's capability to implement an ERP.

2 Literature analysis

At the beginning of the 1990's, ERP (Enterprise Resource Planning) are the heart of the information systems of most of the companies. The investments made in this type of IT are often motivated by the rationalisation allowed by the integration of processes (Markus, 2000). These applications respond to a certain number of inconveniences discovered during the evaluation of information systems and originating principally from their heterogeneous and patchwork construction (Reix, 2004): problems related to the communication of data from different domains, to the difficulty of reaching synthesis status, to the high maintenance costs due to the heterogeneous nature of the application fleet or even to the difficulty involved in training users to the different application environments proposed. Among the advantages usually ignored regarding the ERP's, we can consider several elements that are involved in the logic of rationalization of the company information system and it's organization: the conception of a unique database, the deployment of standard processes through different company functions, the availability of reliable monitoring indicators, etc.

In management science, the literature on ERP can be divided into two types: one concentrated on conditions, *ex ante*, associated with ERP project success and failure factors and the other concentrating on the induced effects (Robey *et al.*, 2002). In our case, we consider that multiple exogenous and endogenous elements are likely to put the different project stages at risk (Markus, 2000). Based on the study of about 40 articles on the subject, we have identified no less than 22 different risk factors.

The case of the ERP's is probably one of the most complete illustrations regarding changes induced by the deployment of an information system. Their functional model consists into adopting and applying "business best practices" considered for the sector in which the firm competes. In other words, with an ERP, priority is given to the implementation of "standard modules" instead of specific developments. In that way, the adaptation of the application to the particularities of the firm is confined to the configuration of those modules (Davenport, 1998). A form of inflexibility is therefore put forward and some authors even go so far as to consider this operation mode as a new form of Taylorism (Gilbert & Leclair, 2004).

Leaving company particularities in the background and implementing pre-established process models is not without potential problems of resistance by employees. These problems can be specific to some individuals or may even grow into inter-personnel or inter-group conflicts (Barki & Hartwick, 2001). The efficiency of an ERP deployment project includes therefore a strong social character that must not be neglected for the risk of failure (Besson, 1999). Our analysis of the literature makes us identify two levels of resistance for the participants of ERP implementation: the first concerns the operational dimension and the second concerns social and political dimensions (Table 1). Hereafter, we will present the elements that compose these two dimensions.

1) *The conflicts regarding the definition of task execution.* Some frictions may arise in determining "best practices" that will be adopted when configuring the ERP (examples: the manner of establishing invoices or orders, data collection and coding of articles, the validation process of internal documents). Robey *et al.* (2002) reveals that the problem of the ERP is less based on the capacity of the company to manage change but more on the difficulty of users to understand the way they were supposed to complete their tasks. For Besson (1999), these conflicts may first raise internally in the form of the different processes confrontation or the same task involving several participants of the company. They can also appear externally by the alignment of the company policy with the "best practices" considered with the ERP implementation. In that case, conflicts are often illustrated by users claiming for a better consideration of the specificities of their needs (Besson, 1999; Markus & Tanis 2000; Larif & Lesorbe, 2004) and roundabout usages of ERP functions (Davenport, 1998; Markus & Tanis, 2000; Gilbert & Leclair, 2004).

2) *The competencies conflicts deals with* the expertise required for the completion of the task rather than the manner in which this will be completed. The job of accounting manager is one of the classic illustrations of professional changes induced by an integrated information system. While the job included, up to the appearance of the ERP, an intense work of collection, aggregation, synthesis of accounting and financial data, the job have moved to automated tasks supported by the ERP. The reduction of low added value tasks has led accounting manager to be more focused on analysis and consulting tasks.

3) *Value conflicts* concern the perceived objectives assigned in terms of value creation (Aubert *et al.*, 2002). For instance, in the hospital sector new processes imposed by the ERP were perceived like a market takeover attempt inconsistent with a public service mission (Besson, 1999; Ménard C. & Bernier C. (2004). In private sector companies this type of conflict may appear in different forms because of the firm finality profession. Among these, we could talk about the well-known "Moscow eye" to which these types of IT are often compared by users.

Table 1. Conflicts related to the implementation of an ERP

	Type of conflict	Examples of associated work
Operation dimension	Conflicts about the definition and the execution of tasks that the users must fulfill.	Robey et al., 2002; Markus & Tanis, 2000; Larif & Lesorbe, 2004
	Conflicts about the new professional skills	Robey et al., 2002; Markus & Tanis, 2000; Newman & Westrup, 2005
Socio-political dimension	Conflicts of values	Robey et al., 2002; Aubert et al., 2002; Menard & Bernier, 2004
	Conflict due to a loss of power	Hart & Saunders 1997; Watson et al., 1999; Jasperson et al., 2002.; Bancroft-Truner & Morley, 2002

4) *Power conflicts* concern the distribution of autonomy and the influence capability of participants. On one hand, ERP's can give key users more power by providing them real-time data processing ability (Davenport, 1998). On the other hand, they can increase the transversality of tasks and then reduce the independence of employees (Markus, 1983). As highlighted by Gilbert & Leclair (2004), in a classic management system, the individuals are not generally forced to give the information required by their collaborators, but they produce their own individual work. The organisation induced by the ERP brings a higher number of mutual instructions. The integration of the Information System thus represents a management vehicle of interdependencies (Rockart & Short, 1989) by which the user prescribes the conditions and the means of their colleagues.

Given the organizational upheavals induced by the deployment of an ERP, lot of research projects agree that a project of this type can not be correctly completed with few implication of the general management. The hierarchy must be a "sponsor" (Davenport, 1998; Markus & Tanis, 2000) of the project and make its "publicity" by assorting credible objectives (Goodman & Sproull, 1990). However, according to the threats perceived by the implementation of an Information System, the users may use their influence power to promote or even to inhibit the project (Beaudry & Pinsonneault, 2005). Indeed, power should not be perceived only as a vehicle of formal decision granted by an authority or by withhelding resources. At the heart of an organization every actor has a reasonably large margin of freedom to gradually increase their power (Crozier & Friedberg, 1977). Those sociopolitical games illustrate the management problem induced by information systems.

Therefore, it is not surprising to observe that the experience of a company in the area of information technology project management is considered as a key competency (Wilcocks & Griffths, 1994; Akkermans & Van Helden, 2002) in the deployment of a sophisticated information system like ERP's. However, for Crozier & Friedberg (1977), key skills existing in an enterprise may also represent action enablers in such power games. In this case, individuals can use their recognized experiences as a vector of influence and as a force of persuasion to orient choices in the way of their personal interests. Consequently, the issue regarding the ambivalent character of the enterprise expertise in the domain of IT must be raised. To what extent can this expertise be used as positive enablers for the deployment of an ERP?

3 Case-study

NETIA, a French SME (located near Montpellier), is one of the leaders in broadcasting (40 countries covered). Its customers are TV channels and public radios like, BBC, ABC, Rai uno, Canal+, France Télévision, etc. Created in 1993, the company employs 70 persons spread over two sites in France and subsidiaries abroad (Amsterdam, Liege, Rome and New York). The firm is an IT service agency dealing with the deployment of audio and video data digital solutions. Besides development, its activity consists of implementation management (consulting, process analyse, engineering, training, maintenance and evolution).

The information system of Netia has been developed progressively by ad-hoc initiatives and requirements. These isolated and independent developments have involved a lack of data coherence as well as an excessive growth in the number of applications required to treat these developments. Consequently, a large part of the employee tasks was used to provide data in order to feed all of the parallel systems installed to response to local needs. For example, the control service has developed a set of Excel programs to partially deal with a divided use of the SAGE software used for the accountancy. Each process (the arrival of an order form, a delivery form, etc) corresponds to a data entry for one or more shared Excel files (on the server there is a file for the order forms, another for the clients, another for prospects, etc.) The operational structure of the information system consists therefore of Microsoft office files from which the data is manually extracted in order to produce management indicators required for the company control. Thus, the loss in productivity becomes apparent not only in the multiple repeated data entries due to the absence of information integration, but also by redundant procedures attempting to ensure a type of reliability by the systematic and repetitive cross referencing of data related to operations. The lack of integration of the information system is also highlighted by data access problems. Thus, the project coordinator can not know the status of the client order in progress without contacting directly the logistics service who must consult the SAGE application in order to respond. Given that the transaction history is dispersed throughout several isolated management applications, purchase tracking (in the case of client feedback or use of a guarantee) is difficult to retrace. Client invoicing is not automatically triggered by a delivery. Logistic managers must enter the information in an Excel file shared with the account department in order to know the state of the process, etc.

It is thus the ensemble of the administrative personnel who was asking of the deployment of an integrated information system to ensure a more coherent and efficient management of the tasks. We can at this point highlight the originality of this case, where the project is not requested by the management but directly by the users who are usually described in documentation about the ERPs as potential resistance vectors.

Two failed attempts at the integration of an information system have been made at Nétia. In 1998, a project to install an ERP was launched. The coordinator in charge of the project carried out an initial study that lasted more than six months and ended by the abort of the project. The second try have concerned a CRM application which was planned to be purchased in 2002. Tainted by an initial feeling of failure, this second attempt was also abandoned.

4 Research methodology

If Netia is expert in IT project management, from a client point of view, it turned out to be unable to apply its expertise to itself. Different reasons incited us to adopt an "action-research" methodology for this purpose:

- Because of the aborted previous projects, the company was eager of recommendations about IS project management from researcher point of view.
- The SME had a short budget concerning this project and could not afford buying the services of a consulting agency.
- In information systems research, "action-research" is turning out to be a more popular and accepted methodology (Baskerville & Myers, 2004).

Our interventions began the first semester of 2005 and the first step conceived with top managers was to identity the explicit and tacit conflicts explaining the ERP project rejection. We conducted 8 semi-directive interviews spread out over four months.

We identified the existing subcultures and examine in which ways these drove the company into dead-end situations. The interviews were divided among representative employees of the different professions (see Table 2).

The interview grid used (see appendix) has been conceived with reference to the risk factor lists of Markus *et al.* (2000), Akkermans & Van Helden (2002), Besson *et al.* (2002). To avoid some reluctance the interviews were realised in a one-to-one interaction and with an anonymous format of the responses collected. During a first part of the interviews the employee was asked to select on the grid, the factors he considered as explaining the rejection of the project. In a second part, we asked him to explain what happened and to develop his perceived dissents and tensions that occurred between employees for the ERP project. Each interview lasted around one and a half hour.

Table 2. Interviewees conducted

Initials	Service	Function
VB	Accounting	Management coordinator
AG	Computer	Computer Dept. Coordinator

	Dept.	
PV	Computer Dept.	Software developer
SR	Accounting	Supplier invoicing
SB	Accounting	Client invoicing, salaries
OC	Operations	Project Director
PD	Logistics	Logistics coordinator
XZ	Sales	Sales coordinator

5 Results and discussion

The interviews conducted allowed us to identify a deep opposition between the IT staff and the administrative staff (management control and logistics in particular) toward the ERP project.

5.1 A conflict of values

During the interviews, the developers put forward "the inconsistency of ERPs to the needs of Nétia" as the official reason of their reluctance. Nevertheless, other interviews allowed us to transcript the following declarations that highlight a conflict of values:

- Project coordinator statement: *"my analysis on the lack of evolution and integration of the Information System is: the IT staff is professional regarding computer based applications. So, they develop the tools they like and need without worrying about coherence. Thus we could not impose the development of others collaborative systems despite the overwhelming number of meetings!"*
- Management controller statement: *"When they* (the developers) *examined the interface and the application functions they were systematically pessimistic: I would have done better than that, in my opinion it's not great!"*
- Developer statement: *"I prefer open source tools"*.

So, here the conflict of values concerns the adoption of a "ready to use application" like an ERP instead of pushing for an internally developed solution. In this context, it seems that the strong IT culture works against the acquisition of professional applications offered on the market. One of the management controllers said: *« If the developers were not at the heart of the company activity, we would not have had these problems. »* We can now draw links to this conflict of values between developers and user with the work of Ballé & Peaucelle (1972) which exposed that the culture of developers often collides with the logic of manager. Because managers are more concentrated on the completion of their work rather than on how the applications used to this purpose are constructed or should be constructed.

However, a second level of analysis reveals that this apparent situation hides a power conflict.

5.2 A power conflict

Developers represent a key competence asset for Nétia. Effectively, the broadcast software's developed by the company are not standard or straightforward applications that can be bought from a classical software editor. Composed by solutions invoiced for several K€, these programs ensure the storage, the management and broadcasting of audio and video programs. Therefore, very specific skills are required regarding sound, image and storage (on servers of several Terabytes), broadcasting by satellite, etc.

The developers in the company represent a reasonably rare workforce on the market and this gives them some power with regards to the hierarchy. Thus, they gained, overtime, strong independence in the completion of their tasks. *« I decide my own objectives! »* declared one of the developer coordinators interviewed. An administration coordinator described for us the holiday management example: *"The developers were use to freely organize their work depending on the tasks and on the assignments to be completed. They do not really respect the procedures for taking holidays. Holidays are taken without prior booking. Instead of filling out the relevant forms and having them validated by the hierarchy, the requests (when they are made) usually take the form of an informal conversation"*. However, the implementation of an ERP implies the deployment of formal processes that are inconsistent with this type of *ad hoc* processes. Regarded as a control tool, such system represents a threat for the developer's independence that they gained.

For their part, the top managers have, until now, been passive and they have avoided imposing this unpopular solution to the computer personal. An administration coordinator stated: *"If we really wanted to impose a standard solution, we could. However, this would mean interfering with the developers. But they are the makers of the programs sold, so…"* Moreover, the fact that there has been no concrete or major prejudice due to the unreliability of the information system does not particularly motivate top managers to discuss more this situation. *«Regarding the successful implementation, the management favours the R&D, only the R&D. . . the rest, such as improving organization, is not considered as vital. »* The developer coordinator told us his feelings according to which this lack of attitude by the top management could be explained by *« a lack of awareness of the necessity to modernize the information systems »*. However, the same person declared not doing all the necessary in order to attract the attention of management for this point.

6 Conclusion

Nearly all of the work done on the resistance to change resulting from the possible deployment of an ERP was concentrated on the opposition of users to the operating mode of acquiring the ERP. Moreover, it is often considered important to involve the « key users » in the project team from the beginning. An important point about the case study is that the main inhibiting factor was not the users themselves (in this case they were asking for the ERP) but the developers who were opposed to the idea of using a package application made by another company. Certainly perceived as a potential threat to their independence, the preference for a "ready to use application" against a software editor could also be interpreted as raising doubts regarding their expertise. Even if the present information system results in reduced productivity, these losses appear not to be judged sufficient by management to proceed with the implementation of a new system to which the developer staff are opposed and thus risk social climate degradation.

The scientific and professional literature on IS is enthusiastic about work distinguishing *risk factor* and *success factors.* If those models could be useful in terms of project management, it is however convenient to avoid all determinist approaches and to adopt more contingent analysis that may reveal that IT expertise could in fact become weakness and vice versa. That's why the dual perspective on the firm's capability to ERP implementation is important to consider. In fact, the dual perspective opposes the IS research perspective that supports the IT experience as a success factor and our case study perspective which considers the IT experience as a risk factor. In IS research, most of the articles on ERP are considering the lack of IT experience as a *risk factor.* However, the case study highlights the organizational maturity in the IT field is not necessarily a *key success factor.* Adopting an ERP equally requires (perhaps above all) an experience in terms of information system outsourcing.

This article equally highlights the risks linked to a management style avoiding conflict management (Barki & Hartwick, 2001). If change must be achieved, then the coordinators must accept the inevitability of certain conflicts, try to channel expertise, and even impose arbitrary choices, rather than letting the organisation sinking into the abyss of power and influence games.

References

Akkermans H., Van Helden K. (2002), "Vicious and virtuous cycles in ERP implementation: a case study of interrelations between critical success factors", European Journal of Information Systems, vol. 11

Aubert B. A., Rivard S., Patry M. (2002), "Managing IT outsourcing risk: lessons learned", in R. Hirscheimm, A. Heinzl, J. Dibberns, Information Systems Outsourcing, Enduring Themes, Emergent Patterns and Future Directions, Springer, pp. 155-176

Ballé C., Peaucelle J.-L. (1972), Le pouvoir informatique dans l'entreprise, Les Éditions d'Organisation.

Bancroft-Truner & Morley (2002), "Organisational Politics is about to go positive", Training Journal, August

Baskerville R., Myers M. D. (2004), “Special issue on action research in information systems: Making IS research relevant to practice – foreword”, MIS Quarterly, vol. 28, n°3

Barki H., Hartwick J. (2001), “Interpersonal Conflicts and Its Management in Information System Development”, MIS Quarterly, vol. 25, n°2

Barki H., Rivard S., Talbot J. (1993), “Toward an Assessment of Software Development Risk”, Journal of Management Information Systems, vol. 10, n°2

Besson P. (1999), “Les ERP à l’épreuve de l’organisation”, Systèmes d’Information et Management, vol. 4, n°4

Besson P. (1999), “Les ERP à l’épreuve de l’organisation”, Systèmes d’Information et Management, vol. 4, n°4

Beaudry A., Pinsonneault A. (2005), “Understanding User Responses to Information Technology: A Coping Model of User Adaptation”, MIS Quarterly, vol. 29, n°3

Crozier M., Friedberg E. (1977), L’acteur et le système : les contraintes de l’action collective », Edition du Seuil

Davenport T. H. (1998), “Putting the Enterprise into the Enterprise System”, Harvard Business Review, July – August

Gilbert P., Leclair P. (2004), « Les systèmes de gestion intégrés. Une modernité en trompe l’œil ? », Sciences de la société, n° 61

Grover V., Jeong S., R. Kettinger W. J., Teng J. T. C. (1995), “The implementation of Business Process Reengineering”, Journal of Management Information Systems, vol. 12, n°1

Hart P. & Saunders C. (1997), “Power and Trust: Critical Factors in the Adoption and Use of Electronic Data Interchange”, Organizational Science, vol. 8, n°1

Jasperson J. S., Carte T. A., Saunders C. S. (2002), “Power and Information Technology Research: A Metatriangulation Review”, MIS Quarterly, vol. 26, n°4

Lapointe L., Rivard S. (2005), “A Multilevel Model Of Resistance To Information Technology Implementation”, MIS Quarterly, vol. 29, n°3

Markus M. L. (1983), “Power, Politics and MIS Implementation”, Communications of the ACM, vol. 26, n°6, June

Markus M. L., Axline S., Petrie D., Tanis C. (2000), “Learning from adopters’ experiences with ERP: problems encountered and success achieved”, Journal of Information Technology, vol. 15, n°4

Markus M. L., Tanis C. (2000), “The enterprise system experience: from adoption to success”, in R. W. Zmud, Framing the domains of IT management: projecting the future through the past, Pinnaflex Educationnal Resources

Menard C. & Bernier C. (2004), « Le cas d’une mise en oeuvre ERP réussie aux centres hospitaliers de LaSalle et de Verdun: comprendre la démarche par laquelle s’installent les nouvelles façons de faire », Gestion, vol. 28, n°4

Newman M., Westrup C. (2005), “Making ERPs work: accountants and the introduction of ERP systems”, European Journal of Information Systems, vol. 14, n°3

Reix R. (2004), Systèmes d’information et management des organisations, 4ème édition, Vuibert

Robey D., Ross J. W., Boudreau M.-C. (2002), "Learning to Implement Enterprise Systems: An Exploratory Study of the Dialectics of Change", Journal of Management Information Systems, vol. 19, n° 1

Rockart J. F. & Short J. E. (1989), "IT in the 1990s: Managing Organizational Interdependence", Sloan Management Review, vol. 30, n°2

Rowe (1999), « Cohérence, intégration informationnelle et changement: esquisse d'un programme de recherche à partir des Progiciels Intégrés de Gestion », Systèmes d'Information et Management, vol. 4, n°4

Watson E. E., Schneider H., Ourso E. J. (1999), "Using ERP Systems in Education", Communications of the Association for Information Systems, vol. 1, n°9

Willcocks L., Griffiths C. (1994), "Predicting risk of failure in large scale IT projects", Technology forecasting and social change, vol. 47, n°2

Appendix: Key factor grid

Project size	Number of users outside of the organization Number of users inside the organization Number of people in the implementation team Team diversity Number of hierarchy levels occupied by users Number of business units concerned
Lack of internal expertise in project management	Lack of expertise in the implementation of information management plan Lack of expertise in information technology organization Lack of experience and expertise in the contract management related organization Lack of inter-functional representation in the team Dependence on users
Organizational context	Insufficient resources Conflict intensity Lack of clarity in the definition of roles Organizational complexity and level of geographic dispersion Level of inter-service cooperation Level of the functional specialization Level of vertical centralization of decision taking Lack of engagement by the project team Lack of engagement by the higher management levels Level of organizational growth

An Analysis of Communication Technologies for Distributed Embedded Systems in Industrial Process Automation

Pavel Rusakov and Dmitry Mikoyelov

dmitrijs@klase.lv

1 Introduction

With the help of researchers of single-chip microprocessors, it is now possible to manufacture very small embedded devices and gather these in one network. In most cases, the network consists of various sensors, which communicate with the data gathering nodes, data processing nodes, data storage nodes and data visualization nodes (in case a human-machine interface is used to monitor such system). In general, each of such components can be a subsystem of a distributed system.

It is a fact that the current technologies quickly transform into outdated or obsolete ones. This inspires the development of new devices, sensors and systems, to research and manufacture a vast amount of new solutions every year. It is obvious that a distributed industrial process management and control system, which we are researching in this paper, cannot exist without outside connections. This means that we have to design such system having sensors and subsystems situated in different remote locations or vital parts of the industrial process. The problem of communication between these subsystems and the central system is raised.

Wired communications tend to bring a lot of problems and obstacles in building a flexible and fast-response system. Wireless network of subsystems tends to be the most flexible solution in most of the cases. In this paper, we are analyzing different wireless communication technologies, providing examples of implementations and present our conclusions on benefits and drawbacks of each of the modern technologies.

2 The characteristic features of communication technology analysis

In order to objectively compare the technologies used to build communication links between the remote subsystems of a distributed system, we have used the following values (Low, Moderate, High, and Very High {and Worldwide as an extremely high factor for range of service}):

1. **Reliability.** A factor that specifies the reliability of the system implementing the selected technology: the stability of the communication links, the guarantee of delivery of data, error-correction possibilities, etc.
2. **Adaptability.** A factor that specifies the possibility to modify or change (also improve) the configuration of the distributed subsystem network according to the new demands of the system.
3. **Scalability.** A factor that specifies the possibility to enlarge the quantity of subsystem in the distributed system without significant changes of the system structure, configuration or additional expenses.
4. **Complexity.** A factor that specifies the overall complexity of the system implementing the selected technology: how hard is to implement each layer of technology, including middleware communicating devices and similar.
5. **Costs.** A factor that specifies the cost of implementation of the system using the selected technology, which consists of two main components: installation costs, including all the hardware, software and middleware, and running costs (per-message, per-megabyte, per-minute) if applicable.
6. **Range.** A factor that specifies, how far can a distributed system span in space, if it is based on the selected technology. This factor is working with the natural ranges of service for the devices implementing the selected technology.

3 Wireless LAN technology

3.1 Advantage/disadvantage analysis

A wireless LAN is a local area network that uses electromagnetic waves (radio or infrared waves) as its carrier. It has a wireless access point, to give a network connection to all users in the surrounding area. Areas may range from a single room to an entire campus. The backbone network usually uses cables, with one or more wireless access points connecting the wireless users to the wired network [5]. There are various versions of WLAN standard IEEE 802.11: (Wi-Fi), an alternative ATM-like 5 GHz standardized technology, and faster standards: 54 Mbit/s 802.11a (5 GHz) and 802.11g (2.4 GHz) [5], [6], [7].

Wireless technologies can bring many benefits to industrial applications, one of them being the ability to reduce machine setup times by avoiding cabling. So far, however, wireless technologies have not gained widespread acceptance on the factory floor. One reason for this lack of acceptance is the difficulty in achieving the timely and successful transmission of packets over error-prone wireless channels.

In an industrial or factory floor setting, for example, the benefits of using wireless technologies are manifold. The cost and time needed for the installation and maintenance of the large number of cables (normally required) can be substantially reduced, thus making plant setup and reconfiguration easier. In terms of plant flexibility, stationary systems can be wirelessly coupled to any mobile subsystems or mobile robots that may exist in order to achieve a connectivity that would otherwise be impossible.

3.2 Possible schemes of subsystem communications

The easiest way to communicate two distributed system components (or subsystems) is a well-known *point-to-point* method. Here, two subsystems initiate a bidirectional connection. The second possibility is *point-to-multipoint* method. This approach has one base station or access point, which controls communication with all of the other wireless nodes in the distributed system.

In industrial settings, it can be hard to find a location for an access point that provides dependable communication with each endpoint. Moving an access point to improve communication with one endpoint will often degrade communication with other endpoints [11]. There is a major drawback in this solution, because it may be possible to wire together multiple access points in order to improve reliability, but the cost of additional wiring can defeat the original reasons for choosing a wireless solution.

The third possibility is *peer-to-peer method*. A subsystem can send and receive data frames, but in this kind of network organization, it functions as a router and can relay data for its neighbors. A packet of wireless data will find its way to its designated destination, passing through intermediate nodes with reliable communication links.

3.3 Use cases (case studies of possible system structures)

Let us review a water treatment facility as an example of successful implementation of WLAN industrial solution in building a distributed embedded data-acquiring device system. The example is based on Ember Corporation experience (Fig. 1).

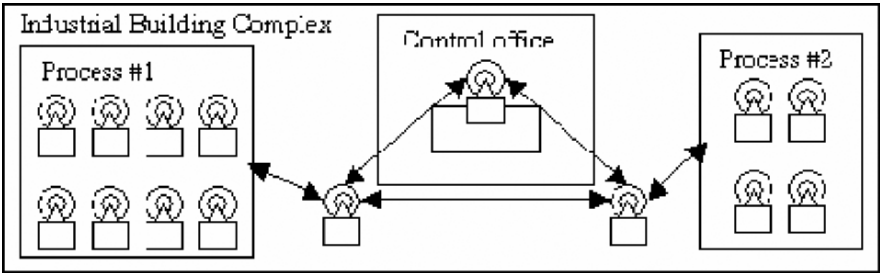


Fig. 1. Use case of Wireless LAN peer-to-peer method

Let us review the analysis of the system integration case:

- **Environment.** The environment includes significant wireless environment hurdles: thick reinforced walls, segmenting giant tanks of water with a lot of metal pipes between tanks.
- **The goal.** The goal is to connect the sensors in the pipe groups back to the control panel located in the control room on the third floor of the water filtration plant.
- **Benefits.** RS-485 serial bus wired to each sensor is replaced with wire-independent solution. Subsystems self-configured on power-up and began attempting to send data to the control room. Relaying subsystems between process parts improve the strength of signal.
- **Drawbacks.** Expensive middleware is needed in each subsystem due to relaying hardware. WLAN demands other technology if the parts of the system are deployed at significant distances. Using huge amount of relaying nodes is very inefficient.

4 GSM and GPRS technologies

4.1 Advantage/disadvantage analysis

The Global System for Mobile communications (GSM) is the most popular standard for mobile phones in the world. GSM phones are used by over a billion people across more than 200 countries [2]. GSM differs significantly from its predecessors in that both signaling and speech channels are digital.

GSM is a cellular network, which means that mobile phones connect to it by searching for cells in the immediate vicinity. GSM networks operate at various different radio frequencies: 900MHz and/or 1800MHz (USA and Canada: 850MHz and/or 1900MHz) [2], [4]. The technical fundamentals of the GSM system were defined in 1987. In 1998, the 3rd Generation Partnership Project (3GPP) was formed [3], [4].

General Packet Radio Service (GPRS) is a mobile data service available to users of GSM mobile phones. It provides moderate speed data transfer, by using unused TDMA channels in the GSM network. GPRS is integrated into GSM standards releases starting with Release 97 and onwards. First it was standardised by ETSI but now that effort has been handed onto the 3GPP [2]. The total available bandwidth can be immediately dedicated to those users who are actually sending at any given moment, providing higher utilisation where users only send or receive data intermittently [9]. GPRS originally supported (in theory) IP, PPP and X.25 connections [8].

GSM DATA is a technology that allows two subsystems to communicate bidirectionally. This approach needs the successful handshake operation to be performed between two GSM modems, until a tunnel between target subsystems will be made available for the information stream. The operation process of this approach is similar to that applied to analogue modems [1], [2]. The main benefit of communication links based on the GSM DATA is that large amounts of data can be transferred in both directions without limitations on session time or data amount. Unfortunately, one subsystem cannot communicate with more than one of remote subsystems. However, it is possible to make an operational environment for each of the distributed subsystems, which will close the connection to the remote subsystem after finishing the data transfer and initiate a new connection to the other remote subsystem. This approach is not suitable for distributed embedded systems with time-critical requirements to the initiation frequency of the inter-subsystem communication sessions.

The second GSM-based technology is *GSM SMS* (Short Message Service), which differs from GSM DATA or GPRS. SMS approach does not require the mobile entity including the subsystem to be active or to be within network range - messages will be held for a number of days until the phone is active and gets within the range. The main benefit of GSM SMS is that short messages have a validity period, which means that it is not expected to be delivered immediately. The message can wait in a mobile operator queue until receiving system can handle it. This approach is useful for such distributed systems that have remote subsystems, which are not constantly available in the network range. Unfortunately, time-critical systems do not allow this kind of lags in data delivery time. The second drawback is that the data sending may take up to 5 seconds. To override the sending and response timeout it is good to use a separate microcontroller for communication routines. Thus, SMS is not recommended in time-critical systems. Fortunately, most of the cases with gathered data transfers do not require extremely quick reactions and general process timing.

The *GSM GPRS* (General Packet Radio Service) technology significantly differs from other GSM technologies. Internet is used to transfer data, whereas the GPRS is the data protocol used for communications between remote subsystems. The GPRS offers high-speed data transfer from 56 up to 114 Kbps and continuous connection to the Internet for remote subsystems of the distributed embedded system. However, this technology brings serious difficulties in the inter-subsystem communication structure. For example, GPRS communication method, which needs an IP address for every subsystem, gives a possibility to connect to any other remote IP address, but impossible to connect to the system with mobile operator local IP address. It is possible to get a corporate IP (not random local) address range from the mobile operator. The second solution is presented in the next chapter.

4.2 Possible schemes of subsystem communications

Let us analyze the design of communication network between remote subsystems with GSM DATA. A possible implementation scheme of the method is based on authors' research in [8], [9], [10] and [12].

In case of implementation of GSM DATA approach, a *supervisor and monitoring* node should have a pool of GSM modems to support numerous connections simultaneously. In fact, this is a significantly weak point of GSM DATA approach. This drawback disappears in SMS or GPRS. We will analyze these issues later in the paper.

Next, let us proceed to the design of communication network between remote subsystems with GSM SMS. The scheme of possible implementation is similar to Fig. 3. However, GSM SMS approach brings significant benefits over GSM DATA, but still has its own drawbacks. For instance, an amount of data sent in a single data-message is limited to 140 bytes. But in the same time, GSM operator central is buffering all the messages and retries the delivery unless the *validity period* is not reached. Also, the central is informing if the message was delivered. Moreover, GSM module collects the received messages in special memory slots, until these can be processed and deleted. There is no way of receiving an erroneous message, meanwhile in GSM DATA the error-correction mechanisms must be implemented. Let us analyze a possible scheme of a data-acquisition subsystem whose task is to send requests for current data and receive responses from remote data-gathering subsystems (Fig. 2).

Fig. 2. Remote subsystem communication with GSM GPRS

There is a solution for building a connection between remote subsystems with the problem of dynamically assigned random IP addresses. The method involves an additional subsystem, which is a server with a real IP address, used to handle each session of the remote subsystem communication. The server is used to receive requests from remote subsystems, store and route the responses to the corresponding subsystem. Thus, the problem of data redirection is solved.

4.3 Use cases (case studies of possible system structures)

Let us overview the proposed distributed system, which is using embedded communication middleware in order to share current state of subsystems with the main monitoring station (Fig. 3).

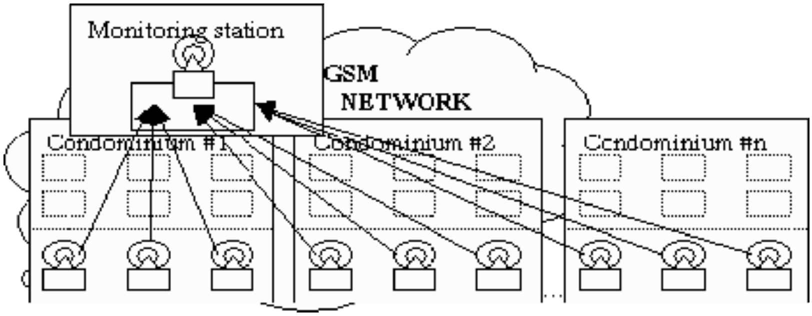


Fig. 3. GSM SMS case study: Monitoring and control of subsystems

The target of the system is automated centralized data acquiring from water flow meters (sensors; may be also electricity flow meters) located in each entrance of each condominium connected to the network of this distributed system. Low cost embedded devices (based on [1], but enriched with analogue to digital converters (ADC) in order to digitalize the analogue data coming from original sensors) are implemented in each subsystem.

- Gathering subsystem: water sensors connected to one data-collecting subsystems with an addition of middleware that sends several packages of data to the central monitoring station (one device for each entrance of the condominium).
- Monitoring subsystem: sends requests to remote data-collecting subsystems and gathers stores response (data-processing algorithms can be applied on the stored data).

In this case, we have a following distributed system:

- **Environment.** The environment consists of the main monitoring station and numerous condominiums scattered remotely around the main station. Each condominium has entrances serving all its water flow meters.
- **The goal.** The goal is to connect all the subsystems (condominiums) to the main station to organize a centralized stet monitoring and control of consumers' flow meters.
- **Benefits.** Cheap grouping of nearby entrances to one subsystem and linking these to the main station. Ability to monitor and control remote subsystems via the main station. Connection is spawned on demand, minimizing sending costs.

- **Drawbacks.** Connection to the main station requires subsequent charges for each short message. The charges are low, but contiguous. Message consists of a limited number of bytes (140).

For GPRS, the target distributed embedded system consists of remote subsystems. Consider that subsystems are situated in an industrial environment, where common communication methods cannot be applied: rough and heavily used terrain, high buildings and metallic or shielded constructions between the subsystems. Cables cannot be used, as well as direct infrared or similar links since serious obstacle or interference. GPRS greatly increases the flexibility of the distributed embedded system adding mobility to each subsystem. This means that each subsystem is not tied to its initial placeholder. This brings the ability of a subsystem to change its location continuously, without reconfiguration of communication. Moving objects (containers, robots). Thus, implementation of GSM/GPRS sets new grounds in process automation.

Let us consider system of several lighthouses, each using a generator, which charges accumulators used by the lamp. The generator has an electric starter and other controls, which allow the controlling subsystem to switch it on/off and read some state information. There is a system supervision server, which shows current states of the subsystems and sensor data. Also, a supervisor with a mobile phone can remotely monitor the system and change states of the subsystems by sending commands. There are two subsystems: “temperature monitoring” and “heating control”, connected directly (close disposition). The remaining subsystems are communicating using GSM GPRS method, sending control messages only when it is necessary.

Let us review the analysis of the following system integration case:

- **Environment.** A distributed system consisting of a generator, which supplies power for charging accumulators used by some power consumer. Two subsystems, “temperature monitoring” and “heating control”, are connected directly due to close disposition.
- **The goal.** The goal of the automation case is to connect all the subsystems (multiple lighthouses) to the main station to organize a centralized stet monitoring and control of each node.
- **Benefits.** Subsystems are communicating using GSM GPRS method, where the connection is permanent, ready to send data any time on demand. Data sending costs are applied only to the amounts of sent data. Ability to monitor, control, and charge remote subsystems via the main station interface. Connection is spawned only when needed, minimizing sending costs.
- **Drawbacks.** The charges for sending are very low due to small amounts of data, but still considerable.

5 Comparative analysis

The research involved the investigation of some selected scientific literature sources, personal research and contiguous practice in the field of automation of industrial processes and implementation of distributed embedded systems. The everyday research allowed gathering all the experience in one piece and covering as the topic of this paper. As GSM based solution in conjunction with wired network for local devices in the constraints of a single subsystem of a distributed embedded system (wired solution implementation depends on the given environment too much and thus is very case-specific; it is almost impossible to conclude on one perfect technology) resulted as the most flexible and effective of all the competitors, a set of universal middleware subsystem devices was established with a design of the first prototype (described in [8], [9], [10], [12] and [13]) of an embedded GSM SMS/GPRS based hardware/software solution to solve many cases in industrial process automation and beyond.

The analysis procedures performed in previous sections allow us to introduce the results of a comparison of the technologies in a form of two logically correlated tables (Table 1 and Table 2).

Table 1. Reliability, Adaptability, Scalability

Method	Reliability	Adaptability	Scalability
WLAN Point-to-Point	High	Low	None
WLAN Point-to-Multipoint	Low	Low	Moderate
WLAN Peer-to-Peer	High	High	Yes
GSM DATA	High	High	Moderate
GSM SMS	Very High	High	Yes
GPRS	High	High	Yes
GPRS (corporate)	Very High	Very High	Yes
GPRS SMS	Very High	High	Yes

Table 2. Complexity, Costs, Range

Method	Complexity	Costs	Range
WLAN Point-to-Point	Low	Low	Low (local)
WLAN Point-to-Multipoint	Moderate	Low	Low (local)
WLAN Peer-to-Peer	High	High	Low (local)
GSM DATA	Low	Moderate	Worldwide
GSM SMS	Low	Moderate	Worldwide
GPRS	Low	Moderate	Worldwide
GPRS (corporate)	Low	High	Worldwide
GPRS SMS	Low	Moderate	Worldwide

General outcome:

- GSM and GPRS solutions are the best choice where the industrial process profit can handle low, but still contiguous running costs (GSM: short messages, data calls; GPRS: data transmission).
- WLAN solutions are suitable for building industrial process automation distributed systems, where running costs are a crucial factor. Thus, system integrators have to avoid the obstacles that have to be faced in the process of implementation of these technologies. Nevertheless the Bluetooth technology is similar to WLAN, the possibility of its implementation in industrial distributed systems is very weak for now.

6 Conclusions

The main goal of industrial process automation is centralized monitoring and control centre. This increases the number of remote subsystems, which need communication middleware. The second goal of industrial process automation is reduction of industrial system maintenance costs. This includes getting the current status, updating firmware, and making changes to the action sequence the subsystem performs.

While some of our proposals (i.e. GSM-based solutions) require low, but obvious and contiguous running costs, these approaches bring benefits of flexible data gathering and secure connections. Meanwhile, approaches that do not require any running costs for data transmission (i.e. WLAN, Bluetooth) reduce the flexibility and bring a sensible instability of communication links or require static subsystem positions (movement from the installation point is very limited due to wire lengths) and high installation costs.

The systems implementing more than one technology are the best solution, where we implement different approaches, which suit each specific bottleneck of the distributed system. For example, local station sensors can be connected using WLAN, but each remote station can be connected with all the others using GPRS connections. At the same time, such systems require numerous middleware subsystems for interfacing different communication protocols and handle the integrity of the data in each point of the distributed system. Thus, hybrid solutions require significant time and resource consumption.

This work has been partly supported by the European Social Fund within the National Programme “Support for the carrying out doctoral study programm’s and post-doctoral researches” project “Support for the development of doctoral studies at Riga Technical University.

References

1. 3GPP Consortium (1999) Network Architecture TS 23.002, http://www.3gpp.org/
2. ETSI (2005) The original GSM standard (GSM 07.05, GSM 07.07, GSM 03.40, GSM 03.38), http://www.etsi.org/
3. GSM Association (2005) GSM technology overview, http://www.gsmworld.com
4. GSM technical overview (2005) http://radio-electronics.com
5. IEEE 802.11 (1999) The original 1 Mbit/s and 2 Mbit/s, 2.4 GHz RF and IR standard
6. IEEE 802.11g (2003) 54 Mbit/s, 2.4 GHz standard (backwards compatible with b)
7. IEEE 802.11h (2004) Spectrum Managed 802.11a (5 GHz) for European compatibility
8. Mikoyelov D (2003) Microcontroller solutions for industrial process automation (Master's thesis), Riga Technical University
9. Mikoyelov D (2003) The development of operation environment for AVR microcontrollers (Engineer's thesis), Riga Technical University
10. Mikoyelov D (2006) An Implementation of Modern Communication Models in Process Automation, 40th Spring International Conference on Modelling and Simulation of Systems (MOSIS'06), Prerov, Czech Republic
11. Poor R, Hodges B (2002) Reliable Wireless Networks for Industrial Systems, Boston, USA, Ember Corporation
12. Rusakov P, Mikoyelov D (2005) Subsystem Communication in a Distributed Embedded System, 13th International Conference on Information Systems Development, Lithuania
13. Rusakov P, Mikoyelov D (2006) An Analysis of Distributed Embedded Systems in Industrial Process Automation, Design, Analysis, and Simulation of Distributed Systems 2006, Alabama, USA

How to Identify Objectives and Genres in E-Democracy Projects: Learning from an Action Case Study

Øystein Sæbø

Department of Information Systems, School of Management,
Agder University College, Kristiansand, Norway
Fax: +4738141029, Phone: +4738141626, E-mail: Oystein.Sabo@hia.no

Abstract: An increased number of public organisations engage in E-Democracy projects to improve their capability to communicate on democratic issues. Such efforts are complex due to lack of knowledge on how to develop information technology solutions to support the complex nature of the electronic communication taking place. In this paper a process is proposed, identifying objectives and genres in E-Democracy projects. The process addresses two major problems identified from the E-Democracy literature and a case study. Firstly, the purposes of E-Democracy projects are often unclear and somewhat naïvely understood. Secondly, it seemed difficult to enact technology to achieve the identified objectives. This paper first describes the suggested process and then focus on experiences from an action case study. The opportunity to link main ideas (phase 1) and genres (phase 2) showed importance in the discussion about what to develop in the action case project. Introducing E-Democracy models simplified a comparison between alternatives and initiated a discussion on the objectives before focusing directly on technology, which is found to be a weakness in other E-Democracy projects.

1 Introduction

E-Democracy refers to the use of information and communication technology (ICT) in political debates and decision-making processes, complementing or contrasting traditional means of communication. The idea of democracy leans fundamentally on effective communication and informed decision-making about public issues among citizens, politicians, officers and other stakeholders who may relate to the decisions (Habermas, 1996; Van Dijk, 2000). There is a need to address the connection between ICT and the electronic communication taking place in E-Democracy projects in more detail (Smith, 2000; Steyaert, 2000).

Experiences from a case study (Rose & Sæbø, 2005) identified two challenges related to the use of ICT in E-Democracy projects. First, the purpose of the E-Democracy project was poorly understood and not shared among major stakeholders in the project; the main objectives were not clear. Second, the connection between the objectives and the use of ICT were unclear. To address these challenges a process for identifying objectives and genres in E-Democracy projects was suggested (Sæbø, 2006). Introducing four E-Democracy models (Päivärinta & Sæbø, 2006) permits different expectations, motivations, and interests to be identified and investigated. The next challenge — how to enact the objectives to information technology — is addressed by linking knowledge on E-Democracy Models, Genre of communication and IT artefacts (Päivärinta & Sæbø, 2006). In this paper the process is first briefly described. Then the paper presents experiences from an action case study, allowing learning and reflection on the suggested process.

2 Theoretical background

The first problem, how to identify main objectives in E-Democracy projects (Rose & Sæbø, 2005), arose from analyses based on a Democracy Model framework (Bellamy, 2000). A democracy model describes a stereotypical form of democracy and outlines how it operates in practice. Literature on democratic models (Held, 1996; Lively, 1975; Van Dijk, 2000) uses varying characteristics to clarify differences among democratic ideas, making a detailed comparison of the competing models difficult. A review of this literature conducted by Päivärinta and Sæbø (2006) suggest a simplified comparison of various E-Democracy models based on two fundamental characteristics: inclusion in decisions and control of the agenda (Dahl, 1989). The four E-Democracy models are introduced in table 1.

Table 1. Models of E-Democracy (based on Päivärinta and Sæbø (2006))

	Partisan E-Democracy	Direct E-Democracy
Citizens set the agenda	Citizens express bottom-up opinions and critique on existing power structures. No explicit connection to the existing governmental or political decision-making processes is defined beforehand. Citizens set the agenda for public discussions but not for decision-making. ICT is introduced to obtain visibility for alternative political expressions uninterrupted by the political elite.	Citizens participate directly in decision-making processes. The citizens online affect the decisions to be made (mostly at the local level). Citizens set the agenda both for public discussion and decision-making. ICT is a crucial pre-condition for democracy to support coordination among decision-makers.

	Liberal E-Democracy	Deliberative E-Democracy
Government (politicians and officers) set(s) the agenda	Government serves citizens who participate in elections and related debates. Government would like to inform and be informed by the citizens without a clear connection to the decision-making process. ICT is introduced to improve the amount and quality of information exchange between government and citizens.	E-Democracy projects are used for targeted purposes involving citizens in the public decision-making processes. The citizens have a good reason to expect that their voices are being heard concerning a particular matter. ICT is developed for increased citizen participation and involvement in the decision-making processes.
	Citizens mainly implicitly included in decision-making processes.	Citizens have an explicitly defined role in decision-making processes.

The second problem, how to link objectives to ICT, is closely related to the first. Main objectives for E-Democracy projects can be identified from the suggested E-Democracy models. Genre theory is one way of studying the emergence of new media or sub-media (Ihlström, 2004) and is introduced here to explore detailed viewpoints on communication patterns for E-Democracy purposes. Finally, knowledge on information technology needs to be more explicit to the specific technology needed. Theories on IT-artefacts focus on the technology and its connection to tasks, structures and contexts. Knowledge of IT artefacts is connected to knowledge of genres and E-Democracy models and is used to explain the link between main objectives and ICT. A review based on the link between these three strands of research (Sæbø & Päivärinta, 2006) identified technological forms for E-Democracy models. These are introduced in table 2.

Table 2. Communication genres for different democracy models

Form	Substance
Partisan Democracy	
Discussion forum	Channel for expressing opinions with little or no visibility under the prevailing political system
Chat system	Synchronous system for short messages
Information Portals	Provide either information on a particular view or as much neutral information as possible

Newsgroups/Usenet groups	Asynchronous discussions, allow longer threads when messages are not in real time
Mail-based discussions	Asynchronous, introducing push-technology by sending mail to participants
Weblogs	Broadcast a citizen's view
Liberal Democracy	
Discussion forums	Information exchange among stakeholders without a clear connection to decision-making
Dialogue system	Citizens express their views as input to decisions made by politicians
Information broadcasting	Bring information from politicians to citizens
Governmental homepages	Inform citizens about timely issues
E-Debates between candidates	Broadcast debates between politicians
Information portals	One-stop access point for information achievements
Consultation	Government/politicians respond to citizen's questions
Candidate or campaigning websites	Promote a candidate or a case
Weblogs	Broadcast a politician's view
Deliberative Democracy	
Discussion forum (issue-based), E-Docket	Initiating, drafting, and defining political issues
Dialogue system	Citizens express suggestions and ideas of issues
Invitation to submit suggestions	Citizens submit suggestions
(e-) Referendum	Inform decision-makers about citizens' view on a particular issue
Homepages	Inform and educate citizens about timely issues
On-line transmissions of meetings	Broadcast meeting for more transparent decision-making
Citizen panel/"jury"	Getting information from a sample of citizens concerning an issue
On-line questionnaire/Survey	Getting opinions from citizens on particular issue

E-voting/Membership ballot	Getting opinions from citizens/members of a community on particular issues
"Your question"	Citizens ask questions to politicians
Public opinion messages	Citizens express their opinions
Real-time chat, Group-to-group chat	Citizens and politicians discuss issues
Closed discussion forum	Party members can affect opinion within a party
Expert panel	Choosing appropriate background documentation for a targeted debate
Formal consultation report	Collecting viewpoints from targeted debate for decision-makers
Feedback about targeted discussions	Informing discussion participants how the discussion affects the decisions
Direct Democracy	
User Registration	To get rights to act in the community
Open discussion/idea forum	Citizens raise new issues and discuss them
Decision-making on issues to be debated	Decide which issues are to be debated and voted on further
Targeted debate forums	Discuss issues proposed for formal discussion
Background documentation	Inform users about timely issues and decisions taken
E-Voting	Decide how to act
Information about the party	FAQ, history, organization

2 A process for identifying objectives and genres in E-Democracy projects

The suggested process (Sæbø, 2006) has two major phases and addresses the criticised approach of concentrating on technology first without identifying strategies and purposes (Grönlund, 2003; Olsson, Sandstrom, & Dahlgren, 2003; Ranerup, 2000; Tops, Horrocks, & Hoff, 2000). The first phase concentrates on identifying objectives for the forthcoming projects. The second phase concentrates on how to enact technology to meet the identified objectives. Phase 1 results in an overview on what democracy model(s) to support. The identified genres (table 2) act as a starting point for the discussion on how to enact technology. Table 3 presents suggested activities for the process.

Table 3. Steps, participants, outcomes and suggested tools for the process

Steps	Participants	Outcomes	Suggested tools	Relation to theory
Phase 1: Identifying the purpose of the project				
Identifying major stakeholders	Project initiators	An overview of stakeholders to include in the process.	Interviews Mapping techniques Workshops	Precondition, no direct connection to the theories involved.
Analysing objectives and purposes	Stakeholders	Stakeholder's objectives are identified.	Interviews Workshops Surveys	Individuals' objectives are identified according to the four democracy models (table 1)
Consensus-building on main objectives	Stakeholders	Common understanding of objectives in the project.	Workshops Scenario building Interviews	Objectives are agreed on supporting one (or several) of the democracy models (table 1).
Phase 2: Enacting identified purposes into suggested genres				
Steps	Participants	Outcomes	Suggested tools	Relation to theory
Identifying technological opportunities	Stakeholders	An overview of different opportunities and reflection on the usefulness of different alternatives.	Prototyping Pilot testing Workshops Interviews	Technological opportunities are identified according to the genres (table 2) for the specified democracy model(s)

Developing a list of objectives and genres.	Project owners	Prioritized list objectives and potential genres to guide the forthcoming development process.	Workshops Interviews	An overview presenting the democracy models to support (table 1) connected to suggested genres (table 2)

3 Exploring the suggested process in an action case study

To explore the process in a real-life context, an action case study was conducted in Kristiansand, a local municipality in Norway. The municipality decided to focus on the "Internet as a facilitator for increased political participation", and describes themselves as a municipality focusing explicitly on openness and dialogue.

Braa and Vidgen (1995) characterise action cases as "action components [that] reflect the potential for research to change organizations, resulting in changes to the social world. The case component reflects the understanding of findings in an organizational context". Action case studies are characterised by: short duration time, interventions in real-time, inclusion of case study elements to support understanding of the domain, emphasis on small (quasi)-experiments in real life-settings, reduced complexity, and focus on changes in a small scale (Braa, 1995). Thus the action case approach makes a good candidate for exploring the suggested process in a small-scale study.

The data sources include dialogues with major stakeholders, project documents, e-mail correspondences, and minutes from project meetings. Ten persons indicated by the executive officer in the project as holding key roles related to the forthcoming E-Democracy project were investigated. Six of them were politicians and four employees in the public administration.

4 Results

The intervention took the following two phases (table 4).

Table 4. Steps conducted in the action case study

Steps	Participants	Outcomes	Tools	Relation to theory	Comments
Phase 1: Identifying the purpose of the project					
Identifying major stakeholders	Stakehold appointed by the executive officer.	Twelve stakeholders identified: six politicians and six staff personnel.	E-mail correspond-ence Dialogue (by the executisve officer	*Precondition, no direct connection to the theories involved.*	Two of the staff personnel were not available for involvement in the project.
Analysing objectives	10 stakeholders	Stakeholder's objectives identified.	Dialogue	Support for Liberal and Deliberative democracy models identified	
Consensus-building on objectives	*10 stakeholders*	*(Not achieved in the action case)*	*No activity took place*	*No activity took place*	*No plenary activities took place, so the consensus building was mainly ignored.*
Phase 2: Enacting identified purposes into suggested technological forms					
Identifying technological opportunities	10 Stakeholders	Individuals' ideas on the use and usefulness of different technological opportunities	Dialogue	Technologi-cal forms supporting Liberal and Deliberative democracy models identified.	Opportunities discussed with each individual.

Developing a list of objectives and potential genres	Researcher	Temporary list of opportunities	Researcher's analysis	Main objectives and suggested genres presented (table 6).	The task is conducted by the researcher involved

4.1 Analysis phase 1: Identifying the objectives for the project

The first step — identifying major stakeholders — was conducted by the executive officer in the project. She appointed stakeholders based on their role in the project, earlier experiences on E-Democracy projects, and their availability for participation in the research project. The second step, analysing objectives, focused on the four different democracy models presented in table 1. The discussion surrounding the models was significant as participants began to reflect on the purposes of the project, but also on advantages and challenges for the different democracy models.

Discussing the Partisan Democracy model participants commented:

- Partisan democracy is not for us; it is not the municipality's concern.
- It is easy to get into an educator-role and disrupt the idea by becoming the initiator.
- You'll risk stealing the show if the municipality is interrupting (in Partisan democracies)…. If the municipality interacts, don't you end up in another model?
- Is this the politician's concern? Isn't the main concern for the press and media?
- In Partisan democracy different organizations or stakeholder groups have the main responsibility. But I can't work as the responsible editor in the municipality if the responsibility is given away in that respect.
- Those who would like to go into action have to do it themselves. And everyone is able to develop a web-page if needed.

Key learning. The participants did not consider supporting Partisan democracy as the municipality's main responsibility. The participants are also worried about interrupting a free political discussion. The model as such is not seen as irrelevant, but it should be developed and maintained by actors outside the municipality, such as media and other stakeholder groups.

On the Direct democracy model participants commented:

- If you had small responsible units with money to spend, you might have had direct or deliberative democracy. But I don't believe in it, the representative democracy still needs to be the main model.
- In direct democracy, single subjects will obtain too much space. You'll lose the comprehensive overview needed in a democracy.

- Referendums require clear answers, yes or no or at least clear alternatives. And that is not the case in our society.
- A local area might vote on a specific topic, I don't disagree with that. In practice, on rare occasions a topic affects only a defined local area.
- Referendum is very difficult, almost impossible, to organise traditionally. It has to be on rare occasions and on specific topics.
- … (Direct democracy) costs a lot of money. And better decision-making is not to be achieved either.
- Referendum by Internet comes to a new opportunity. Maybe we can perform referendums easier, by voting for or against road tolls. But I don't accept a referendum for or against immigration, which is based on values I don't discuss. It is more a fight against selfishness we all have and is therefore more difficult.
- Referendums should be performed on major principal subjects. Electors have to know the opportunities and live with the consequences. By referendums, the people's will is represented, then politicians have to arrange their action according to it.

Key learning. The Direct democracy model achieved minor support. A missing tradition for referendums and high costs are seen as obstacles for this model. Referendums are not considered (by most of the participants) to interact sufficiently with the complexity needed in a modern democracy. The Internet may decrease the costs needed to perform referendums, but it does not change the challenge of achieving involvement by citizens.

On the Liberal democracy model, participants commented:

- Liberal democracy is unproblematic. It's a question on getting information in and out; the challenge then is the quality control on the information.
- I wish we had a huge element of Liberal as well as Deliberative democracy.
- I'm not sure if the decisions must always, at least in superior cases, be made by the county council. It is the only agency having the opportunity to keep a holistic view on the municipality, being willing to make painful decisions prioritizing one subject over another. I can't really see how that changes by the influence of Deliberative democracy.
- By the end of the day, politicians are making the decisions. Citizens are only adding contributions in the processes.
- The citizens' main contribution is to add good advice…. If there are many contributions concerning a subject, it's a sign of the importance of that subject, it's like taking the heat on citizens' concerns.
- I really don't know what a discussion forum should be except for securing publicity on single topics. It would have been interesting if some politicians stepped forward and said what they meant, but also what they wasn't sure of — please come and influence me! — that someone really asked to be influenced. But then they really have to be serious, to let themselves be influenced. The

problem in politics is that opinions are rarely individual; they are commonly decided for a party group.

- What's important for me as a decision-maker is to get as many contributions as possible, a wide range of viewpoints so that I can sort out the best and decide what to use later in the process.

Key learning. Liberal Democracy achieved support from many participants. The Liberal democratic idea of achieving more information without influencing the way decision-making is performed is seen as unproblematic by most of the participants. No promises are made on some kind of direct influence by citizens' participation. The Liberal model combines the opportunity to get input with the traditional representative democracy and is therefore seen by many as the only realistic opportunity.

On the Deliberative democracy model, participants commented:

- We need to strive for Deliberative democracy where it is possible. I can't see any other opportunity on the decreasing participation we now explore.
- I believe politicians would like to stay in the Liberal quadrant, being able to claim a comprehensive communication with the target group, but still making all the decisions themselves. That would not be very popular. To succeed, I think we need to get to the two models here (Direct and Deliberative Democracy). People don't want to engage without any influence. Then we fool them.
- I would like to see a shift from Liberal democracy towards more Deliberative democracy, in any case in a municipality like Kristiansand working exactly on such challenges.
- I think we should work more in the direction of Deliberative democracy, where it is possible. I can't see any other opportunity except the absence of Democracy, which may be present in general in our society.
- I would like politicians to give away some power on single topics, where it is possible. Sometimes you ask on stages where the opportunity to influence is absent, where inputs are only a finery. Then I think it's better not to ask.
- We have to go for the Deliberative model in the future. The party politics engage fewer and fewer people, so we have to develop new ways for citizens to influence and maintain the democracy.
- In an ideal world, the Deliberative model is the one to develop. But politicians are seeking power, that's why they become politicians. They would like to be seen as democratic, listening to others and so on, but I think there are some stable decision-oriented structures in the politicians that are difficult to change.

Key learning. The Deliberative democracy is by many seemed as an ideal model to support. Two main issues were discussed. Firstly, the tension between the model and the politicians' will to really be influenced by citizens is highlighted. If politicians are not willing to be influenced, solutions supporting Deliberative democracy should not be developed. Secondly, Deliberative democracy is by some seen as the only way to engage citizens. Citizens would like more than just the opportunity to speak; they are considered to look for some real influence.

The third step — Consensus-building on objectives and purposes — was not achieved in the action case study. The action case study faced restricted resources that did not allow consensus-building activities as part of the first stage, as suggested in the process (Table 3). The second stage is therefore based on individual views of how to enact technological forms with different models of democracy.

4.2 Analysis phase 2: Enacting objectives to E-Democracy genres

The analysis in stage 1 identified support the Liberal and the Deliberative E-Democracy models. On identifying technological opportunities in the Liberal democracy model, participants commented:

- You may develop layers of information on a web-page, including history, alternatives, and choices. The problem now is that the information is hard to get because of the complexity and tone in public documents. So technology may be used to introduce different topics more briefly and add links to more information on each subject.
- Instead of raising your hand and asking questions, people could send in their questions via sms', a communication form they actually know. Then it becomes like the TV-channels, with a window on the screen continually including new messages showing what the youth are engaged in.
- We would like to go for radio transmission from the county council meetings… Radio transmission seems to work perfectly well. The representatives do not seem to be interrupted at all. Examples in which politicians are videotaped illustrate how they became stiff in front of a camera.
- We need to develop a question and answer kind of service. But my hypothesis is that sending in an enquiry to a politician or a party office without getting a response feels like a slap in your face.
- We are currently redesigning an information portal. I believe the design of the portal will have great influence on the extent to which people would like to visit the page again.
- Many people retrieve information by e-mail who would not have the information elsewhere. I send out newsletters by e-mail to 30–40 people who, without e-mail, would have no opportunity to receive this information. It's extremely efficient.
- Personally I would like to have a blog where I write some thoughts about what I as a politician am doing right now. That would be great.
- We have discussed the opportunity to broadcast meetings with the opportunity to get instant feedback from citizens by having a computer on the table. I think

it will be too demanding for the elected since you have to be very concentrated to participate in a debate. It will become too intense.
- What is important in a Liberal democracy is the municipality's home page. Personally, I utilise the web to find old subjects and minutes, which I find very convenient.

Key learning. Technological forms for Liberal democracy focus on information exchange. The Internet is seen to enable a simplified presentation of information. Some of the participants find the governance-centric presentation of information to be a major obstacle to citizen participation. The Internet's opportunity to host citizen-centric information channels is therefore important. Other important issues are how to design the web-page, the efficiency of information distribution by e-mail or by blog, and the importance of offering communication channels that are known by the users.

On identifying technology for the Deliberative democracy model, participants commented:

- The only definite idea I have is the opportunity to make a closer connection between the Political Agenda (an archive of minutes and calendar of political meetings) where you'll find all the information needed and a discussion forum where you can take part in a discussion you are interested in. The integration between such services needs to be as tight as possible.
- In our party, we have an internet-part, an intranet-part, a closed internal discussion forum and an extra-net part for the national level. On the local level, we have continually ongoing communication among members, mainly based on e-mail. We would like to have the same communication with citizens, but have not yet either the priority nor the capability to include citizens as well.
- E-based debates between candidates are interesting, so far mostly utilised internally in the party. But they are utilised more and more by committees in the municipality.
- What are needed are simultaneous discussions taking place here and now, allowing for follow-up questions if needed. Without including such a service, our dialogue is useless. Dialogue is here and now. If I add a contribution to a politician, I also expect a quick answer. 10 minutes are ok, there might be more than me contributing, but 48 hours is not ok. For me, the Internet is nearly simultaneous.
- I find chat most convenient for deliberative democracy because you are then able to really discuss.

Key learning. Comments on how to enact Deliberative democracy focused much more on discussions than on information exchange. Participants also commented on the importance of simultaneous communication patterns if real discussions are to be supported.

Table 5 summarise the analysis made based on the action case study. The analysis is based on participants' support for two different models, the Liberal and the Deliberative.

Table 5. Summarized analysis of the action case project

Partisan Democracy	Direct Democracy
Restricted support from the participants. The model is not considered to be a main responsibility for the municipality.	Achieved only restricted support. Challenges such as representativity, costs and complexity of the decisions needed in a democracy are obstacles for utilising a Direct democracy model.
Liberal Democracy	Deliberative Democracy
Supported by participants. Unproblematic link between projects and the traditional democratic system. Quality of the information exchange is seen important.	Supported by participants. Challenging, but also more interesting for citizens than the Liberal model. Opportunities to discuss and influence are seen important.
Technological forms are characterised by ease of access, feedback on questions made, and opportunities for information broadcasting. Examples to evaluate further in the project are: Radio/TV transmission Question-and-answer services Information portals News mail Blogs Quality of the municipality's home-page	Technological forms are characterised by the opportunity to discuss, including feedback mechanisms showing the influence of the contributions, and the opportunity to participate in synchronous discussions. Examples to evaluate further in the project are: Discussion forums E-mail based discussions E-Debates between candidates and citizens Chat

4.3 Participants' reflection on the usefulness of the suggested process

The participants were asked to reflect on the perceived usefulness of the suggested process. They commented:

- I think it is necessary to look at what is needed to succeed. A presentation like this makes it possible to reflect on what is needed instead of just being positive to participate because it (the Internet) is a new way to communicate and therefore worthy by itself.
- I believe the models may bring up a lot of interesting ideas from politicians and act as a starting point for a discussion.
- It is really helpful to point out which direction this leads us, making visible the alternatives

- An overview like this on different opportunities is a nice tool in a decision-making process on where to go; it's a nice systemisation.
- It (the suggested process) is quite solid…. It has to be simplified for practical usefulness.
- It (the suggested process) is currently too complicated, but it helps me as an executive officer to better understand different opportunities.
- The problem is often that we start up with technical solutions before knowing what we would like to achieve, which I found to be meaningless. You need to look at what to achieve, different alternatives and thereafter choose the technical solution. Politicians need to get different opportunities plainly put, including cost accounting and what's expected from them on different alternatives.
- Immediately, it seems interesting to arrange a session for the city council where politicians themselves, based on this process, discuss what they would like to develop, simply a process of increasing awareness on what they like to achieve.
- There's a need for even more concretising of different alternatives. What are you achieving by a discussion forum, by a chat system? I don't know.
- Now we have this tool identifying opportunities by ICT, of course we have to use it (the suggested process) further in the process.

Key Learning. The suggested process was perceived useful by the participants involved in the project. This was underscored by the fact that all participants were able to immediately reflect on different democratic models and technological forms despite only a limited introduction. Participants find the idea of focusing on main purposes before discussing technology to be helpful in the developing process. The presentation was found by some to be too complicated, but despite of this still helpful. Participants address the need for being more concrete about the technology and costs connected to different alternatives.

5 Discussion

The main objectives of the project were identified by introducing the E-Democracy models. The dialogues illustrate the tension between the Liberal and the deliberative models. Some of the participants discussed the difference between the "realistic and unproblematic" Liberal model, and the "wanted, but much more challenging" Deliberative model. The process opens for a further discussion focusing particularly on these two models and their consequences.

The dialogues explored some strengths of the proposed process. Firstly, the immediate opportunity to link main ideas (phase 1) and connected genres (phase 2) showed a promising potential to guide the discussion on what to develop. The participants were able to discuss alternatives and express their viewpoints on what to achieve in the forthcoming project. Secondly, introducing the E-Democracy models simplified a comparison among alternatives and initiated a discussion on the objectives before focusing directly on technology, which is found to be a weakness in other E-Democracy projects (Grönlund, 2003; Marcella, Baxter & Moore, 2002; Rose & Sæbø, 2005).

Some challenges were also identified. The value of the suggested process may increase by being more definite on consequences, especially by estimating expected resources needed by different alternatives, such as time and competence requirements. Such knowledge is lacking because of restricted experience in utilising genres in E-Democracy projects. Further research is needed to explore consequences more in detail to further develop on the suggested process.

The suggested process could be taken into account by practitioners who may want to promote a certain kind of E-Democracy. A practitioner may identify first the assumptions of democracy in the development context in question and then the particular technological forms to be implemented in the system, such as those illustrated by the conducted action case.

By the suggested process, E-Democracy researchers can be specific in relation to the suggested framework whether the target of their research contributes to one particular E-Democracy model or a combination of various models. Furthermore, researchers can be specific when relating new knowledge to the field by identifying genres in light of the process. A new contribution can be identified as a genre instantiation supporting a specific E-Democracy model. Through such analyses, researchers can also inform the future practice of E-Democracy, offering lessons learned in a rather detailed manner.

6 Conclusion

This paper addresses the two identified problems (identifying main objectives and how to connect main objectives to genres) by introducing a two-step process. The conducted action case explored the process and gained insight on strengths and challenges. Supported by participants' reflections on the usefulness of the process I argue that the process shows importance by connecting main objectives and the use of ICT in E-Democracy projects. Participants were immediately empowered to discuss different objectives and connected genres after a brief introduction of the process.

More efforts need to be directed to further develop a dynamic experience base discussing particular E-Democracy genres, allowing for the growth of cumulative knowledge among researchers and practitioners. Particular technological forms should be investigated in more detail to find out more detailed lessons learned, including knowledge of resources needed.

Empirical research on E-Democracy has been a scattered field of experiments, lacking solid theoretical foundations, let alone cumulative knowledge to guide research and practice. The suggested process demonstrates how knowledge from established fields of research shows importance in the immature area of E-Democracy research. More research is needed to further explore and explain the ideas of communication genres when the development on new ICT and communication preferences offer new opportunities for future E-Democracy projects.

References

Bellamy, C. (2000). Modelling electronic democracy, Towards democratic discourses for an information age. In J. Hoff, I. Horrocks & P. Tops (Eds.), Democratic governance and new technology, technologically mediated innovations in political practice in Western Europe. London: Routledge.

Braa, K. (1995). Beyond Formal Quality in Information Systems Design, UiO, Oslo.

Dahl, R. A. (1989). Democracy and its critics. New Haven, Conn.: Yale University Press.

Grönlund, Å. (2003). Emerging electronic infrastructures - Exploring democratic components. Social Science Computer Review, 21(1), 55-72.

Habermas, J. (1996). Between facts and norms: contributions to a discourse theory of law and democracy. Cambridge, Mass.: MIT Press.

Held, D. (1996). Models of Democracy. Oxford: Blackwell.

Ihlström, C. (2004). The Evolution of a New(s) Genre. Unpublished Doctoral Dissertation, Göteborg University, Gothenburg.

Lively, J. (1975). Democracy. Oxford: Blackwell.

Marcella, R., Baxter, G., & Moore, N. (2002). An exploration of the effectiveness for the citizen of Web-based systems of communicating UK parliamentary and devolved assembly information. Journal of Government Information, 29(6), 371-391.

Olsson, T., Sandstrom, H., & Dahlgren, P. (2003). An Information Society for Everyone? Gazette: The International Journal for Communication Studies, 65(4-5), 347-363.

Päivärinta, T., & Sæbø, Ø. (2006). Models of E-Democracy. Communications of the Association for Information Systems, 17, 818-840.

Ranerup, A. (2000). On-line forums as an arena for political discussions. Lecture Notes in Computer Science, 1765, 209-223.

Rose, J., & Sæbø, Ø. (2005). Democracy Squared: designing on-line political communities to accommodate conflicting interests. Scandinavian Journal of Information Systems, 17(2).

Smith, C. (2000). British political parties: continuity and change in the information age. In J. Hoff, P. Tops & I. Horrocks (Eds.), Democratic governance and new technology: technologically mediated innovations in political practice in Western Europe (pp. 71-87). London: Routledge.

Steyaert, J. (2000). Local governments online and the role of the resident - Government shop versus electronic community. Social Science Computer Review, 18(1), 3-16.

Sæbø, Ø. (2006). A process for identifying objectives and technological forms in E-Democracy initiatives. Paper presented at the 12th Americas Conference on Information Systems, Acapulco, Mexico.

Sæbø, Ø., & Päivärinta, T. (2006). Defining the "E" in E-Democracy: a genre lens on IT artefacts. Paper presented at the 29th Information Systems Research Seminar in Scandinavia, Helsingoer, Denmark.

Tops, P., Horrocks, I., & Hoff, J. (2000). Tew technology and democratic renewal: the evidence assessed. In J. Hoff, I. Horrocks & P. Tops (Eds.), Democratic Governance and new technology. London: Routledge.

Van Dijk, J. (2000). Models of democracy and concepts of communication. In K. L. Hacker & J. Van Dijk (Eds.), Digital Democracy, Issues of theory and practice. London: Sage publications.

Replacing a Human Agent by an Automatic Reverse Directory Service

Géza Németh*, Csaba Zainkó*, Géza Kiss*, Gábor Olaszy*,
László Fekete**, Domokos Tóth**

*Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics,
{nemeth, zainko, kgeza, olaszy}@tmit.bme.hu,
**T-Mobil Hungary Rt. {feketel, tothd}@t-mobile.hu

Abstract: Agents who answer the calls in a reverse directory service have to face a considerable challenge: they need to communicate proper names (such as the names of persons, companies and streets). Their pronunciation is frequently irregular and their spelling is not obvious. The authors developed a TTS specialized for this task, i.e. the reading of names and addresses in order to create an automatic reverse directory service. The novelty of our system compared to others developed earlier is that we employed a new reading mode and optimized the acoustic database based on an extensive analysis of Hungarian proper names. This resulted in high intelligibility and naturalness. Our system was launched as a service of T-Mobile Hungary. The specialized TTS can also be the basis of other applications in the future, such as location based services.

1 Introduction

In a reverse directory service, callers usually interact with an agent. After the caller enters a phone number, the agent tells him/her the name and the address corresponding to the number in the directory. In order to automate such a service, a TTS system is needed that is capable of handling most proper names (especially the names of persons, companies and addresses) correctly.

TTS systems developed for this purpose – we call them name and address readers – have been an important and commercially attractive application of synthetic speech. The Orator system, developed at Bellcore in 1990, was the first successful approach to solve this problem for US English [1]. In terms of other languages, Belhoula developed a solution to handle the variety in the structure of names in German in 1993. Unfortunately, it did not become a real-life application [2].

Eloquens, the Italian system developed at CSELT, was first tested by Telecom Italia in 1993 and, after gradual improvements, its quality increased significantly (MOS changed from 2.93 to 3.63) [3]. The main factor of the improvement was the development of a specialized acoustic database and synthesis method for this application and the continuous improvement of text analysis techniques.

For other languages (especially with less speakers), name and address readers have been developed only sporadically. An automatic Swedish reverse directory application was introduced by Telia in 1997 based on extensive text preprocessing (e.g. creation of a lexicon with 200k entries) and an MBROLA diphone acoustic database [4]. Callers were charged only the third of the service fee with human operators.

In Hungary, a TTS-based e-mail reader [5], became a successful service of T-Mobile Hungary. This was the motivation to develop a name and address reader to integrate into the reverse directory service of the mobile operator. Although the quality of the diphone-based PROFIVOX TTS [6] was satisfactory for e-mail reading, higher quality was required for the reverse directory application. The main reason is that e-mail reading has never been available with human operators while getting names and addresses from well-trained agents has been freely available for years (as a replacement for the legal obligation to provide printed versions to subscribers) so callers expect similar quality.

A high-quality name and address reader was developed in a cooperation. The voice of the system had to be the standard female announcer of T-Mobile Hungary and the quality had to be good enough to replace human operators. After successful field trials the system was launched at the beginning of 2004. It is fully automatic, i.e. there is no option to connect with a human operator. After a year of operation, user opinion polls were collected.

2 Structure of names and addresses

Our work was based on a database of app. 3 million wireline subscribers of 1997 out of app. 10 million Hungarian citizens where the association of names and addresses was mixed. A similarly constructed partial database of app. 2 million T-Mobile subscribers was used for validation and cross-checking. With respect to reading, names and addresses may be regarded independent so they will be discussed separately.

2.1 Features of Hungarian names

Hungarian names can be either person names or legal names (e.g. companies, organizations). Directory listings usually contain the whole name in a single record i.e. given, family and legal names as well as labels, titles and other signs might also appear in the record.

Person names may have several forms is Hungarian. In the simplest case there is a family name followed by a given name. There can be more family names and given names (like middle names in English). In case of married women, there are twelve possible forms (based on their and their husbands' family and given names). Placement of titles, such as 'Dr' further complicates the forming of ladies' names.

Hungarian given names are relatively easy to detect as they are selected from an official list of app. 1700 elements, published by the Hungarian Academy of Sciences.

Company names can be recognized from the abbreviation referring to the legal type of the company (e.g. 'Ltd'). In many cases, however, it is not the offical name that appears in the database. The real challenge is the reading of descriptions that do not have a standard way of reading, such as the company name '@rc Kft' (visual correspondence between a and @). To deal with this issue, in our categorization all names that contain other characters than letters and '-' are regarded legal names since they contain characters that are illegal in names of persons.

2.2 Features of Hungarian addresses

Addresses are also given in a single record. Their processing is simpler than names because the structure is relatively straightforward and the order of the elements is fixed (ZIP, city, street name, street type, number). The ZIP code and the city is always given, any of the other components might be missing or replaced (e.g. in case of a P.O. Box).

3 Database categories

Based on the initial analysis of 10k randomly selected database samples and taking into account the rules of Hungarian name creation, 35 name and address element categories have been defined. These define the sorts of elements used to build the name of a person, a legal name or an address. Such element categories are for example the ZIP code or the label for the first part of family names (e.g. 'Dr'). The elements for names of persons are the building blocks of maiden names also (the maiden name can be a part of a married woman's name).

4 Statistical name and address information

The directory contains 2.944.000 records, that were labeled by name and address categories in order to build the database that can provide the input for the TTS. 300.000 records were labeled manually to produce a training and test set. The rest was labeled with our automatic labeling tool. In case of uncertainties, the tool marks corresponding records with a special label. 2.627.000 person names and

285.000 legal names were found. 32.000 records could not be automatically classified. These results represent the first study of Hungarian proper names on such a large scale.

4.1 Given names

The database contained 1797 different given names. 183 of these were not on the Academy list and were regarded as foreign names. It was a surprise to see that the most frequent female name (‘Mária’) is only 23rd on the frequency ranking of all given names and there are only 3 others (‘Éva’, ‘Katalin’ and ‘Erzsébet’) in the most frequent 100. The reason might be the sociologic effect of phone numbers being mostly registered for the name of the husband, and also women frequently using their husband's name instead of their maiden name.

4.2 Family names

The large number of 103.850 different family names of the 2.627.000 proper names found in the database means that the variation of Hungarian family names is significant. Only 90% coverage of Hungarian family names may be reached by the most frequent 20k tokens.

4.3 Name prefixes and labels

Most of the 82 different name prefixes and titles appeared only in a few different positions. Many of them were single characters which have to be clearly separated from the other name elements during reading. This phenomenon has to be taken into account during prosodic planning. The vast majority of titles attached to name components are relatively simple but due to their variable position and possible joint appearance they also have a significant role in the final acoustic and prosodic design of the system.

4.4 Legal names

Approximately ¾ of legal names have a simple structure, suitable for reading. 10% of the examined items contain at least one given name. 10% contain mosaic words. 9% contain numbers and about 2.5% contain special signs.

4.5 Addresses

We compared the results of our address analysis in the directory to data obtained from the Hungarian Postal Service. The Post database lists 3562 different cities and towns while our database search yielded 3523. The number of new names was 29, mainly due to detailed local notation. There were about 16k different street names in the database. They were attached to 77 different street types (75% street, 15% road, 3% square, 7% other).

5 Recommendation for the spelling of Hungarian

During the course of the project it turned out that there is no formal standard of Hungarian spelling. In order to fill this gap a questionnaire of all Hungarian ASCII characters was created and 28 adults (university students and professionals with a university degree) were asked to describe how they would spell the given character. The proposals with the highest frequency were included in the final solution. Most of the reference words are given names (starting by a capital letter) the remaining are well known nouns. Special characters (@, &, #, etc.) are difficult to handle because a large part of the population has never even seen them. They can be regularly found in company names, however.

6 Prosodic analysis and design

Reading names and addresses requires a simpler prosodic structure than general TTS conversion. All sentences are statements. In case of reading the name, the carrier sentence ends with the variable part. The last word in the sentence is always a given name so if the most frequent given names are recorded in a sentence-final position, they can be used in most cases without any prosodic modification. In case of legal names, the name of the unit comes first and the final unit is always a legal name type. This implies that legal name types usually need no prosody adjustment. In case of reading the address, the ZIP code, the city, the street and number together and the further location parameters (e.g. building, floor, door) form separate phrases in the sentence.

7 The acoustic database

Results of the text and prosodic analysis were used to extend the acoustic database of the general purpose Profivox TTS system. The goal was to minimize the need for prosodic (especially pitch) modification and to maximize the use of longer units.

Family, company and street names are highly variable so they are usually generated by the concatenation of diphones and triphones. The most frequent 317 given names (>99% coverage) and all possible legal name types were recorded in a phrase-final position. 82 name prefixes and labels were also recorded in a sentence-final and medial position. In order to synthesize numbers (e.g. ZIP codes) with high quality, the number reader described in [7] was integrated and the number units were recorded. One thousand city names (covering >90% of all subscribers) were also recorded, uttered separately. About one hundred public-place definitions (e.g. street, road, floor, staircase) were also added in both sentence medial and final positions. Units for spelling were included as single statements. Altogether, the acoustic database contains about 12k variable length elements.

By analyzing the synthesis of 30k randomly selected addresses by our system, it turned out that the longer units (e.g. pre-recorded given names, spelling units) and triphones are the frequently applied acoustic elements. Diphones are used only occasionally. Pitch modification is rarely needed due to the careful selection of units and their consistent recording conditions. Prosody modification is usually limited to intensity and timing.

The database compiled for the name and address reading domain can be gradually extended to other domains such as car types, commercial locations, as new services will require.

8 Dialog strategies

Reading of names and addresses in a reverse directory service has to fulfill two contradicting requirements: efficient access to information and intelligible reading (so that the caller can also get to know the proper spelling, if needed).

Looking at the legal name examples of Section 2.1., it is clear that dictating them correctly may not be obvious, even for humans. The following dialogue structure was implemented after an iterative design process:

a) Input of 9 digit phone number by DTMF
b) search for the subscriber of the number in the database
c) if information is not available or secret, notification of the caller by a prerecorded prompt
d) if information is available

1. reading of name after a selection from the three name reading modes:
a) general, continuous ‘overview’ reading mode
b) syllabification mode (a special syllabification algorithm was developed for this purpose [8])
c) spelling mode

2. reading of address in all name reading modes The ZIP code and all other numbers are read by a number-reader [7]. Other elements are read by the specialized TTS.

With this approach, users may first receive a general ‘overview’ of the name and address information and can ‘zoom’ on details as required. The detailed modes provide a way for comprehending proper names efficiently that are irregular in some manner while keeping the menu structure very simple.

9 Evaluation

First, the TTS performance was tested. The testing corpus contained 100k randomly selected and 4k especially important directory records. These units were audibly checked by native listeners. The error rate (improper pronunciation) was less then 3%. Most of the errors were due to rare foreign words.

Initial user testing was also carried out. The general opinion was very positive. It was found that users appreciate the possibility to get detailed descriptions in the syllabification and spelling mode. Asking a human operator to do the same was embarrassing for them. The system was launched as a publicly available service, in fully automatic mode in February 2004.

9.1 User opinion polls after one year of operation

Operational complaints against the system were very limited. Most of the problems were caused by errors in the directory database. During the first 8 months of operation, 63% of the information requests used the overview, 18% the syllabification and 19% the spelling mode. These results show that all modes are exploited.

A formal opinion poll was also conducted by an independent market research company. The initial results reported here are based on the answers of 12 people. The poll is about to be carried out soon on a larger sample.

The subjects were customers who used the reverse directory service recently and were willing to answer questions. Four aspects of the system were rated on a scale from 1 (worst) to 5 (best), with the extremes labeled.

Generally, the users were highly satisfied with the service. The average score for speech quality was 4.58. The speech rate got 4.75, the correctness of the provided information 5.00, the dialog management 4.58. The overall score for the system was 4.92 on average – except one 4, everyone rated the system to 5.

Usage statistics also show a probably high user satisfaction: the number of calls per month did not significantly change since the automatic system replaced the human-based one.

10 Summary

A high-quality name and address reader was developed. The first analysis of names and addresses in Hungary on such a large scale was carried out. As a result, basic statistics about name and address distributions were computed and presented. A recommendation for the spelling of Hungarian was prepared. A syllabification module and a specialized TTS of names and addresses was also developed and a speech-based dialog system was designed. The reverse directory application was successfully introduced into service by T-Mobile Hungary in a fully automatic mode.

Acknowledgements

This research was supported by the Hungarian National Research and Development Program. The authors acknowledge the support of T-Mobile Hungary staff, especially Péter Pósafalvi and János Benke, in the integration of the new system into the existing infrastructure. Tamás Bőhm contributed by reviewing the draft of the manuscript.

References

1. Spiegel, M. F., "Coping with Telephone Directories that Were Never Intended for Synthesis Applications", Proc. of ESCA-NATO/RSG 10 Tutorial and Workshop on Applications of Speech Technology, Lautrach, Germany, 1993, pp. 19-22
2. Belhoula, K., "A Concept for the Synthesis of Names", Proc. of ESCA-NATO/RSG 10 Tutorial and Workshop on Applications of Speech Technology, Lautrach, Germany, 1993, pp. 167-170
3. Nebbia, L., Quazza, S., and Salza, P. L., "A Specialised Speech Synthesis Technique for Application to Automatic Reverse Directory Service", Proc. of IVTTA '98, IEEE-ESCA Workshop on Interactive Voice Technology for Telecommunications Applications, Torino, Italy, 1998, pp. 223-228
4. Lundin, F. J., "The Swedish Automatic Reverse Directory Service", Proc. of IVTTA '98, IEEE-ESCA Workshop on Interactive Voice Technology for Telecommunications Applications, Torino, Italy, Sept, 1998, pp. 219-222
5. Németh, G., Zainkó, Cs., Fekete, L., Olaszy, G., Endrédi, G., Olaszi, P., Kiss, G., Kis, P., "The Design, Implementation and Operation of a Hungarian E-mail Reader", International Journal of Speech Technology, Kluwer, Vol. 3, No. 3/4, December 2000, pp. 217-236
6. Olaszy, G., Németh G., Olaszi, P., Kiss, G., Gordos, G., "PROFIVOX - A Hungarian Professional TTS System for Telecommunications Applications", International Journal of Speech Technology, Kluwer, Vol. 3, No. 3/4, December 2000, pp. 201-216
7. Olaszy, G. and Németh, G., "IVR for Banking and Residential Telephone Subscribers Using Stored Messages Combined with a New Number-to-Speech Synthesis Method", in D. Gardner-Bonneau ed., Human Factors and Interactive Voice Response Systems, Kluwer, 1999, pp. 237-255
8. Fék M., Németh G., Olaszy G., Gordos G.: "Megértést segítő részletező gépi névfelolvasás magyar nyelvre" (Automatic syllabification supporting understanding for Hungarian name-reading), Proc. of 2nd Hungarian Computational Linguistics Conference, Szeged, 2004, pp. 301-306

Topic Identification by the Combination of Fuzzy Thesaurus and Complexity Pursuit

Sándor Szaszkó*, László T. Kóczy*,†

*Budapest University of Technology and Economic, Hungary
†Széchenyi István University, Győr, Hungary

1 Introduction

An information retrieval system allows users to efficiently retrieve documents that are relevant to their current interests. The collection of documents from which the selected ones have to be retrieved might be extremely large and the use of terminology might be inconsistent.

Natural languages use many similar terms for the same or similar concepts. In order to most of the documents belonging to the same topic, special dictionaries have to be set up. A thesaurus is a collection of terms (words) which describe the same concept. With the use of the thesaurus we can discover connections among documents, which do not necessarily contain the same words, or retrieve relevant documents which do not necessarily include any of the query words.

Automated keyword search is the most widespread approach to this problem; however, it is easy to recognise that documents not containing the actual keyword(s), but maybe its their synonyms, or some terms with a closely related but more specific meaning, might be similarly relevant for the search. If the keyword in the query is Soft Computing (SC), documents on Fuzzy Systems, Neural Networks and similar topics will be unambiguously relevant, even if they do not mention the broader term (SC) explicitly a single time. Moreover, other parts of the same scientific community prefer to use the name Computational Intelligence with a rather similar meaning, so all documents related to the latter should be also retrieved.

In previous studies we suggested the use of hierarchical co-occurrence frequencies as indicators of the importance of individual words and groups of words in the contents of given documents 11011. This means that the occurrence frequencies of certain words in the title and sub-titles, the abstract and introduction or conclusion parts most of the documents might be characteristic for the occurrence frequencies of certain (other) words in the main body of the text. The frequency of word A in the title and word B in the text is called their hierarchical co-occurrence.

It is obvious that these frequencies are not probabilistic measures, as it is not the relative frequency of a certain word among all words of the document that directly measures its relevance. However these frequencies determine the possibility degrees of the documents in a somewhat indirect, certainly not linear and essentially non-additive way. In the next section a method for transforming the counted or estimated frequencies of occurrence into possibility measures (fuzzy membership degrees) will be presented.

There are several example for supporting IR by thesauri. 9 examine word coocurring in a 40 word wide window. 7 and 8 construct thesauri by transposing the term-by-document marix. Applying fuzzy logic to automated information retrieval is not new. Some of the most important advances in this field are summarised in 6. In several points of this paper, reference will be made to concepts introduced in this work.

2 Keyword occurrence frequencies and possibility degrees

If analyzing a collection of documents related to a certain topic (e.g. folkloristic beliefs) it will be found that some of the words occur quite frequently in all or most of them, thus these words are of no significance with regards to the contents of any particular document. The words which are common in any natural language document are called stop words, while those, which might be significant in some context but have a role similar to that of the real stop words in a certain context will be called in this study relative stop words. In the context of folkloristic beliefs such a relative stop words are hard to identify. Because of this the set of relative stop words will become empty. These texts are usually rather short and they contain only relevant, often enigmatically short expressions – except the proper stop words.

By the omission of stop words (and relative stop words, if relevant) the set of significant words is obtained which m be used for a further analysis. Some of these words might be more important than the rest and might be chosen as the set of keywords. In a hierarchical co-occurrence approach the titles and sub-titles, etc. might be checked only for keyword occurrences, while the rest of the documents for any significant word. An example for classifying words into these four categories can be seen in Figure 1.

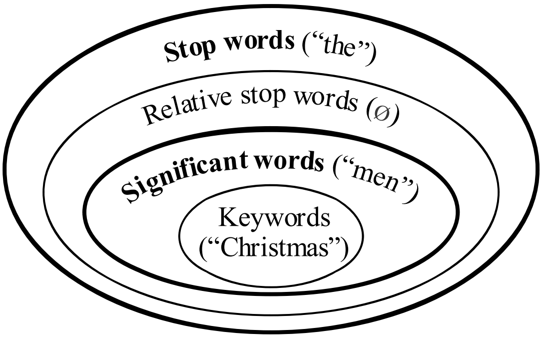


Fig. 1. Categories of words in documents

In the Fig. 1 the four categories of words can be seen: stop word like "the", relative stop words an empty set, and "men" as a general example for a significant word and finally, "Christmas" for a keyword, as beliefs connected to Christmas time form a large and important subgroup of our collection.

It is a crucially important issue how occurrence frequencies can be and are transformed into fuzzy membership degrees, which may be interpreted as important and relevant measures, satisfying the properties of possibilistic measures.

The following must be considered here. Membership degrees or fuzzy measures range from 0 to 1, where 0 expresses the total lack of importance, and 1 stands for absolute importance. Words occurring in a document very frequently are usually stop words (absolute or relative ones), and so they should be left out of consideration. For the remaining class of significant words it is generally true that higher occurrence frequencies indicate higher importance degrees as well. Although the connection between occurrence frequency (word count) and importance degree is strictly monotonic, it is certainly not proportional.

The critical domain is somewhere what can be defined as "a few occurrences", depending on the type and size of the document, generally somewhere between 2 and 20 word counts. It does not matter much whether a word occurs in a document 10 or 12 times, it is likely that this document will be rather important from the point of view of the query in both cases. On the other hand, one or two occurrences of a word might be coincidental or might indicate that the subject is touched upon only very superficially, while repeated mentioning (three or four or more) is an indicator that the word in question is an important word from the point of view of the document. With short documents like beliefs and superstitions these numbers might vary. Especially it can never be expected that words occur more than a few (two, tree or four) times.

The mapping from occurrence frequencies or counts to possibilistic membership degrees is thus generally a sigmoid function, with its steep part around the "critical" area of occurrences – the concrete values depending on the expected lengths and types of documents, and the category of environment (title, text, etc.). These sigmoids $\sigma(F)$ have to fulfill the conditions which are given in 5. In practice σ is not necessarily continuously differentiable, but its characteristics should be nevertheless “S-shaped”.

3 Fuzzy Pre-processing of a Folkloristic Corpus

Hungary has a very rich folkloristic tradition and especially in the last century a successful work has been done to research and preserve this heritage. For instance, nowadays many young people learn the traditional dances of villages, in many cities there are parties with folk music and dance. The 3rd CIOFF World Folklorida, the “Olympia of Folk Art” took place in Hungary very recently, an obvious sign of international appreciation of this work.

Quite a few Hungarian cultural anthropologists were collecting beliefs and superstitions mainly in the 20th century. There are about 27 000 documents on paper in the National Museum. Unfortunately the classical techniques of anthropology are not able to process this amount of data and usually studies analyze only 6 to 10 documents. Of the above collection, there exists a digitized database of 2704 Hungarian belief texts, suitable for computerized analysis, which has been processed by principal component analysis12.

In order to distinguish the different dialects one word is spelled in different ways, sometimes even special characters are used to note the pronunciation down. The other problem is the use of old style language, a big part of the vocabulary of the corpus is not in use anymore. So first of all different appearances of the same words had to be collected into pre-process dictionary. In this way we also solved the problem of Hungarian language being an agglutinative language, it puts many different tags at the end of the words.

After all 1837 significant words remain in the pre-process dictionary. Special attention was paid for negation. Hungarian language puts *nem* (not) word before the verb for its negation, so if the software found the *nem* word in the text, it considered it together with the next word and it searched that whole string in the pre-process dictionary.

At this phase of the research we did not try to solve the problem of words with more than one meaning like *fog*, which can mean tooth and also to catch. The context has to be analyzed to select the actual meaning which is quite a complex task.

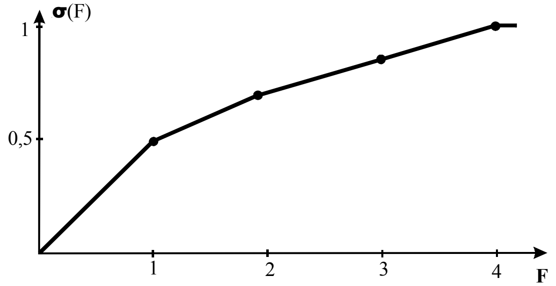


Fig. 2. Sigmoid curves used for the sort documents

The sigmoid function σ(F) used is shown in Figure 2. The documents are usually 2 to 5 lines longs, just very few of them exceed the size of half a page so even a single occurrence is quite significant. Thus in case of a single occurrence the membership degree is already 0.5. Less than 0.1% of the significant words appear more than 4 times, so in case a word occurs for times in a document the membership degree is 1.

By the end of the pre-processing matrix **M** was obtained whose size is 2704x1837, the words (W) and the documents (D) are listed at the edges. The fuzzy degrees stored in **M** show how much a given word is related to a document.

3.1 Word Frequency Degree

Let us define a degree in the following way:

$$I_w = \sum_{i=1}^{N} \sigma_{d_i W} \qquad (1)$$

The Word Frequency Degree shows a given word how significant is in collection of the documents, in the whole corpus.

4 Establishing Fuzzy Pseudo-thesauruses by Co-occurrence

In order to define synonymy take the set of (all) concepts 6. The concepts may be abstract ones, which cannot be found in the real world. Now take the set of words. A word belongs to a concept with a fuzzy membership degree. One word may belong to several concepts and several words may belong to one concept.

Two words are synonyms if they belong to the very same concepts with the same degrees. The fuzzy thesaurus lists not just the synonyms, but also the degree of synonymy. For example if word A and word B belong to some concepts which are common, but they also belong to some different concept they are synonyms with a degree which is less than 1. If A and B are not related to any common concept they are synonyms with the degree of zero.

If we want to obtain this thesaurus there are two main questions:

1st: What to use instead of the set of concepts?

2nd How to choose degree of the synonymy when it is not 0 or 1?

There are not many possible answers for the first question if we want an automatic method to generate the thesaurus. We substitute the set of documents for the set of concepts. Because this is not a very good substitution the thesaurus obtained is called pseudo-thesaurus.

There are plenty of possibilities to define the degree of synonymy between pairs of words. In the next two subsections we will see that the obvious choices do not give good result, so in 4.3 a special weight factor is introduced.

4.1 Establishing the fuzzy pseudo-thesaurus step by step

Step 1: *Co-occurrence degree calculation*

$$\mu'_{ij}(D) = \min(\sigma_{W_i D}, \sigma_{W_j D})$$

$$\mu_{ij} = \frac{1}{C}\frac{1}{s}\sum_{z=1}^{N}\mu'_{ij}(D_z) \tag{2}$$

where C is a constant which keeps μ_{ij} in the range of [0,1], C is independent while s a weight may be dependent from i and j. The first idea how to choose C can be N, the number of the documents, but in this way the values of μ_{ij} are very small. We get more reasonable values if

$$C = \max_{i,j}\left(\frac{1}{s}\sum_{z=1}^{N}\mu'_{ij}(D_z)\right) \quad . \tag{3}$$

Step 2: *Suitable α-cut*

If the number of the significant words is M, then the co-occurrence degrees (μ_{ij}) form a matrix size of MxM, let call it **W**. The words are listed on both sides from 1 to M. Since $\mu_{ij}=\mu_{ji}$ this matrix **W** can be represented by an undirected graph.

Choose a suitable α for which the *α-cut* leaves about 30 to 40 nodes in the graph. This is a representation of a pseudo-thesaurus.

Step 3: *Searching maximal cliques*

An edge means that the connected two nodes representing words are "synonyms" in this broader term ("related" in the meaning). If a set of nodes are fully connected, than they are called a clique, and they are supposed to be related to the same broad concept.

Step 4: *Fuzzy clique*

Many times among the found maximal cliques there are a few which have many common nodes. Since we chose α arbitrarily it is reasonable to check if these close cliques describe the same broad concept and they can be aggregated. We take the cliques which have just one different node and investigate these different nodes. If there is an edge between them on level α'=0.7α cut we aggregate the cliques.

4.2 Weight *s*=1

Here the measure of co-occurrence is simply proportional with the sum of the co-occurrence. Column I_w shows in Table 1 that just very frequent words remained in the α-cut. The most frequent words of the corpus (number 18, "go" and 26, "do") have the most edges.

Table 1. List of words in case weight s=1

NR	HUN.	ENGLISH	I_w
1	ad	giv(e)	74.7
2	asszony	woman, wif(e)	92.1
3	este	evening	103.3
4	fent	Above	69.1
5	fog	tooth/catch	123.5
6	férj	husband	66.1
7	gyermek	Child	160.8
8	György	George	31.8
9	haza	Home	42.8
10	3	3	83.7
11	ház	House	173.6
12	karácsony	Christmas	90.4
13	kicsi	small, little	110.9
14	legény	young man, fel-low, lad	41.7
15	Luca	Lucia (St.)	52.8
16	lány	girl, lass	142.4
17	meghal	die	90.8
18	megy	go, walk	218.4
19	mise	(holy) mass	27.6
20	mond	say	108.3
21	nap	day/Sun	172.3
22	ront	spoil, bewitch	48.7
23	sok	much, many	86.0
24	szent	saint (St.)	37.8
25	tehén	cow	97.1
26	tej	milk	61.3
27	tesz	do, put	177.8
28	tojik	lay eggs	32.3
29	tyúk	hen	77.4
30	víz	water	98.9
31	éjfél	midnight	40.5
32	éjjel	night	87.2
33	év	year	94.7

Even though not all, but many of the edges give really meaningful pairs of words. The maximal cliques show some logic behind, but usually some very frequent words appears as an odd-one-out. (see Figure 3)

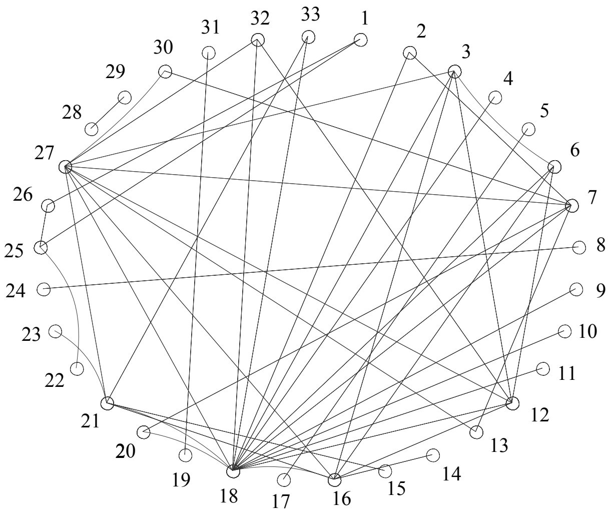


Fig. 3. Graph of a pseudo thesaurus, weight s=1

4.3 Weight s=max(I_{WA}, I_{WB})

To avoid the dominancy of frequent words lets divide the co-occurrence measure, μ_{ij} by word frequency degree (I_w), which is greater from word j and word j. In this case C=1 because I_w is never smaller than the sum of μ'_{ij}-s.

The highest not empty α-cut has 90 nodes. For all words in the α-cut I_w=0.5, which means that all appear just once in the hole corpus. The relations found here have no significance because of the low occurrence.

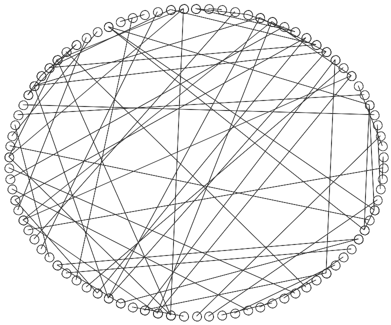


Fig. 4. Graph a of pseudo thesaurus, weight s= max(I_{WA} ,I_{WB})

4.4 Weight *s*=1+ (max(IWA ,I_{WB}))/20

By 4.1. and 4.2. we guess that weight s should be between 1 and max(I_{WA} ,I_{WB}). After several tests weight s=1+ (max(I_{WA} ,I_{WB}))/20 was proved to be the most efficient to identify concepts. As its can be seen in Table 2, these words represent a wider range of I_w values.

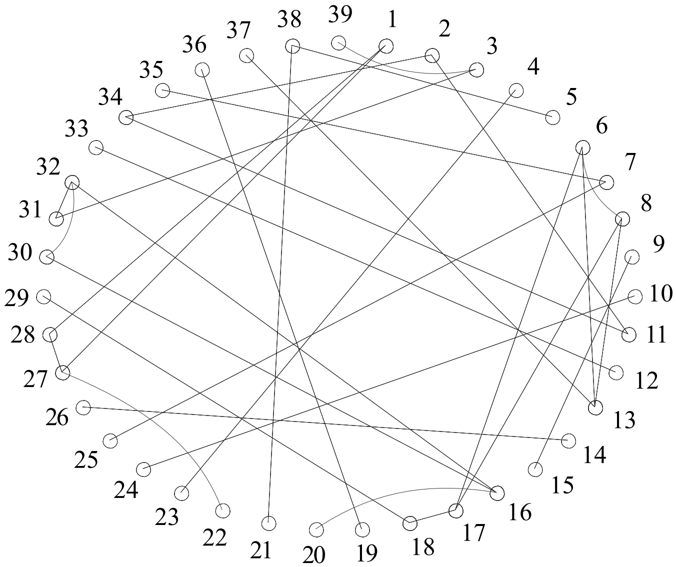


Fig. 5. Graph of a pseudo thesaurus, weight s=1+ (max(I_{WA} ,I_{WB}))/20

Table 2. List of words in case weight s=1+ (max(I_{WA} ,I_{WB}))/20

NR	HUN.	ENGLISH	I_w
1	ad	giv(e)	74.7
2	bal	left	28.4
3	csibe	chicken	35.2
4	csörög	clatter	5.5
5	cédula	peace of paper	9.5
6	este	evening	103.3
7	fecske	swallow	16.8
8	férj	husband	66.1
9	gyermek	child	160.8
10	György	George	31.8
11	jobb	right	30.9
12	jön	come	91.1
13	karácsony	Christmas	90.4
14	kenyér	bread	51.4
15	kicsi	small	110.9
16	Luca	Lucia (St.)	52.8
17	lány	girl, lass	142.4
18	megy	go, walk	218.4
19	mise	(holy) mass	27.6
20	nap	day, sun	172.3
21	név	name	43.3
22	ront	spoil, bewitch	48.7
23	szarka	magpie	6.7
24	szent	saint, St.	37.8
25	szeplő	freckle	10.3
26	süt	bake	33.6
27	tehén	cow	97.1
28	tej	milk	61.3
29	tesz	do, put	177.8
30	tojik	lay eggs	32.3
31	tojás	egg	43.8
32	tyúk	hen	77.4
33	vendég	guest	29.8
34	viszket	itch	25.5
35	Vér	blood	17.6
36	éjfél	midnight	40.5
37	éjjel	night	87.2
38	ír	write	20.7
39	ültet	plant, seat	21.3

The graph in Figure 5 is not dominated by any node, very frequent words like go, do, day (number 18, 29, 20) have not more than two edges. We managed to avoid that some very frequent words are connected to most of the other nodes like in Figure 3. It is logical that words with high are related to more concepts (this is why they are so frequent), but some of them should not be in most of the maximal cliques.

Table 3. Maximal cliques

giv(e) (1)	cow (27)	milk (28)
left (2)	right (11)	itch (34)
evening (6)	husband (8)	Christmas (13)
evening (6)	husband (8)	girl, lass (17)
Lucia (St.) (16)	lay eggs (30)	hen (32)

chicken (3)	egg (31)
chicken (3)	plant, seat (39)
clatter (4)	magpie (23)
peace of paper (5)	write (38)
swallow (7)	freckle (25)
swallow (7)	blood (35)
Child (9)	small (15)
George (10)	Saint, St. (24)
come (12)	guest (33)
charismas (13)	night (37)

bread (14)	bake (26)
Lucia (St.) (16)	day, sun (20)
girl, lass (17)	go, walk (18)
go, walk (18)	do, put (29)
(holy) mass (19)	mindnight (36)
name (21)	write (38)
Spoil, bewitch(22)	cow (27)
egg (31)	hen (32)

Table 3 lists the maximal cliques of the graph of Figure 5, these sets of word describe broad concepts which are important in this corpus. The 3rd an 4th line of the Table 3 differ just in the last word. Take a lover α-cut of Matrix **W**. We can find that already at α’=0.8α there is an edge between Christmas(13) and girl(17), so the two maximal cliques can be aggregated, they form the fuzzy maximal clique: evening(6), husband(8),Christmas(13) and girl(17). It is easy to imagine beliefs which are about how a girl can find a husband at Christmas Eve.

4.5 Two examples:

“Karácsony estélyén doboskát sütnek s a leány az elsôvel kiszalad és amely legénnyel találkozik legelôször az lesz a férje.”

Christmas Eve “doboska” cakes are prepared and the daughter of the house runs out with the first piece and the young man whom she meets first will become her husband.

“Karácsony estélyén a lánynak egy öl fát kell felvenni és ha a számuk páros, akkor férjhez meg, ha páratlan, akkor nem megy.”

Christmas Eve the girls should pick up a bunch of wood, if the number of pieces is even she will be married, if it is odd, she will not be married (namely, next year).

5 Establishing Pseudo-thesauruses by ICA

The idea of using Independent Component Analysis was initiated by 1, there Complexity Pursuit Algorithm was used to identify topics in chat flows. Complexity Pursuit Algorithm is an extended version of Independent Component Analysis, the advantage of CPA is that it is able to separate components from time series while it remains computationally simple.

CPA was tested also on our folkloristic corpus, since theoretically the algorithm could be to able separate important directions efficiently on data which does not contain time information. Experimental tests proved CPA unsuitable for finding meaningful associations on our test corpus, but the parent algorithm ICA showed interesting results. It can be stated and it can be shown that ICA was able to find at least as good words cliques as good ones were found in 1 by CPA.

5.1 Independent Component Analysis (ICA)

Our data model assumes that the observations x(t) are linear mixtures of some latent components:

$$\underline{x} = \underline{\underline{A}}\,\underline{s} \tag{4}$$

where $\underline{x} = [x_1, x_2, ..., x_n]^T$ is the vector of observed random variables. In our case document containing words, $\underline{s} = [s_1, s_2, ..., s_m]$ is the vector of independently predictable latent components, and $\boldsymbol{A}$ is an unknown constant mixing matrix. The statistical independency of s_i is not strictly required.

s_i, the independent component is understood as independent topic in the corpus, so the later found s_i corresponding to set of words. These sets define the topics.

The task is solving equation (4) which has infinite number of solutions. We need to find a possible $\boldsymbol{A}$ matrix in which the entropy of the lines is minimal so it contains the most information. For finding the maximum non-Gaussian directions several comparison function can be used. Further details of the implementation of the algorithm can be found in 3.

5.2 Results

Since ICA uses a deep mathematical approach, the function of statistics in the connection between the resulting set of words and the parameters of the algorithm are quite hidden not like in the case of fuzzy pseudo-thesaurus, where the word frequency on the corpus has high correlation with the role of the word in the fuzzy clique. So in the case of using ICA we have not found special dominancy of some word, meaningful sets contain most of the words of the dictionary.

The following tables show sets of words identifying topics. The lists put together are results of ICA and fuzzy pseudo-thesaurus. The set are ordered in a way that is easy to recognize that the two methods are able to give very similar results.

Table 4. Four identical words in the result of two methods

ICA	Fuzzy
witch	*witch*
peel	*go*
go	*do*
do	*night*
night	water

ICA	Fuzzy
drop	*children*
children	*small*
small	say
do	*do*
water	*water*

Table 5. Three identical words in the result of two methods

ICA	Fuzzy
chick	chick
black	east
armpit	egg
east	
egg	

ICA	Fuzzy
go	give
draw	keep
keep	cow
cow	milk
milk	

ICA	Fuzzy
draw	give
keep	keep
cow	cow
milk	milk
dug	

ICA	Fuzzy
screw	screw
Georhge	dew
Dew	pick
Go	
Pick	

6 Conclusions and further study

Based on some further studies on textual document retrieval a method of automatic thesaurus generation has been proposed and an appropriate weight factor was introduced. The method was applied on a collection of Hungarian folkloristic beliefs. This way a pseudo-thesaurus was generated.

In the way of finding independent components of the terminus x document matrix a set of words were identified. Both kinds of set of words thus found in the thesaurus and ICA analysis were meaningful for the contents of the beliefs analyzed, and helped to understand the course topics of beliefs.

The fact that ICA and the fuzzy pseudo-thesaurus approach give many similar sets of words supposes that the two methods can be combined. We suggest that the topics identified in the fuzzy pseudo-thesaurus should be refined with sets of ICA.

Methods should be tested with different corpuses in order to identify and resolve side effects. In order to improve the usability of the topic identification the introduction of the topic's relevance measure is planned to be introduced.

References

1. E. Bingham, A. Kabán, M. Girolami, (2003) Topic Identification in Dynamical Text by Complexity Pursuit. Neural Networks Research Centre, Helsinki University of Technology
2. K. Chakrabarty, L. T. Kóczy, T. D. Gedeon (1999) Analysis of fuzzy relational charts in information retrieval *IETR99-01*, School of Computer Science and Engineering, University of New South Wales, Sydney
3. A. Hyvarinen (1999) Fast and robust fixed-point algorithms for independent component analysis, EEE Tr. on Neural Nw., 10(3),626–634
4. G. Klir, T. Folger (1998) Fuzzy Sets. Uncertainty and Information, Prentice-Hall, Englewood Cliofs, NJ
5. L. T. Kóczy, T. D. Gedeon and J. A. Kóczy (2002) Fuzzy tolerance relations and relational maps applied to information retrieval *Fuzzy Sets and Systems 126* 49–61
6. S. Miyamoto (1990) Fuzzy Sets in Information Retrieval and Cluster Analysis Kluwer, Dordrecht, 259p
7. Y. Qiu, H. Frei (1993) Comcept Based Query Expansion" SIGIR conf.
8. G Salton, C. Buckley (1990) Improving Retrieval performance by Relevance Feedback Jurnal of A. S. for Information Science, 288-297
9. H. Schütze, J. O. Pedersen (1997) A Cooccurrence-based Thesaurus and Two Application to Information Retrieval. Information Retrieval and Management, Vol. 33, 307-318
10. S. Szaszkó, L. T. Kóczy (2004) Identifying Concept in Folcloristic Corpus by Fuzzy Pseudo-thesaurus *Eurofuse* 2004, Warsaw, 522-532
11. S. Szaszkó, L. T. Kóczy (2004) What Lectures Note About, Identifying Concepts by Fuzzy Pseudo-thesaurus *EESTEC-IEEE Conference*, Italy
12. V. Voigt, M. Preminger, L. Ládi, S Darányi (1999) Automated motif identification in folklore text corpora. Folklore. Electronic Journal of Folklore Vol. 12., Tartu, 1999 126-141 pp. Also available at: http://haldjas.folklore.ee/folklore/vol12/motif.htm

Long-term Preservation of Electronic Information – A study of Seven Swedish Governmental Organizations

Viveca Asproth

Mid Sweden University, SE-83125 Östersund
Viveca.Asproth@miun.se

Abstract: Information technology in combination with changes in organizational structures and forms for work has led to an increasing amount of information and also to totally new forms of information and records creation. Much of the information that earlier was produced on paper is now being produced in electronic form, for example as e-mail and in databases. A 10-year period is a long time when it comes to preserving electronic records. This is of course due to the rapid technological development, for example the problems with obsolete software and file formats. Vitale functions in society such as the social insurance and national registration have to be accessible for long periods of time. Without authentic, accessible, searchable, and reliable records these functions in society are in fact impossible to maintain. There still remain a lot of problems to solve before secure and efficient long-term preservation will be brought about. Digital preservation requires, in addition to the technological development, elaborated strategies, new workflows and organizational structures, standards and common metadata, new specific competences, and close co-operation across different professional fields. Though many problems still remain to be solved, governmental as well as business organizations are in an intermediate stage in the changeover to electronic records and have to deal with this situation. In this paper an empirical study of seven Swedish governmental departments is presented. The study was carried through with help of interviews and in some of the cases examination of complementary documentation. The result of the study shows the stage of changeover organizations are in, and the different strategies they use to handle the mixture of paper-bound documents, scanned documents, electronic records, databases, etc that comprise the information about one single commission or object and which has to be handled both in the short and the long run.

1 Introduction

Information technology in combination with changes in organizational structures and forms for work has led to an increasing amount of information and also to totally new forms of information and records creation. Much of the information that earlier was produced on paper is now being produced in electronic form, for example as e-mail and in databases. The Dutch archive theoretician Eric Ketelaar (1999) has invented the word "archivalisation" to describe the phenomenon that more and more seems to be worth to document and archive. This means that a considerably larger amount of information needs to be handled. Electronically generated and stored information is also exposed to change and manipulation in quite another way than paper-bound information.

A 10-year period is a long time when it comes to preserving electronic records. This is of course due to the rapid technological development, for example the problems with obsolete software and file formats. Goldstein (2004) stresses the need for a quick solution to the problem of archiving electronic information for the future. Further, he says that the major threat to archived material today is not the fire hazard but the rapid development of different file formats for documents, sound and images.

Vitale functions in society such as the social insurance and national registration have to be accessible for long periods of time. Without authentic, accessible, searchable, and reliable records these functions in society are in fact impossible to maintain. The technical development and transition to electronic media has led to that traditional methods have become obsolete or at least insufficient.

There still remain a lot of problems to solve before secure and efficient long-term preservation will be brought about. It seems that although much of the challenges associated with digital preservation is strategic, organizational, and structural and not only technical, the research concentrates at solving the technical issues. Digital preservation requires, in addition to the technological development, elaborated strategies, new workflows and organizational structures, standards and common metadata, new specific competences, and close co-operation across different professions from traditional preservation management to computing science. (Asproth, 2005)

Traditionally, preserving means keeping them unchanged; however, our digital environment has fundamentally changed our concept of preservation requirements. If we hold on to digital information without modifications, accessing the information will become increasingly more difficult, if not impossible.

This situation creates a fundamental paradox for digital preservation: On the one hand, we want to maintain digital information intact as it was created; on the other, we want to access this information dynamically and with the most advanced tools. (Chen, 2001)

Most of the governmental departments strives to become 24/7 agencies. Within 24/7 agencies, the whole idea is that exchange of information is done electronically. The Swedish Agency for Public Management have stated that without possibilities to manage and preserve e-records for long time, the development of working e-government services could be in danger (Statskontoret, 2003). Developing effective e-record management systems is proven to be difficult. For example in electronic healthcare, records have been difficult to access over time, and thereby the preservation issue has been difficult to solve, which makes development of e-record management systems difficult (Grimson, 2001).

Electronic records have proven to be difficult to preserve with maintained reliability and authenticity (Duranti, 2001a; Duranti, 2001b). Electronic records can be spread in separate databases with links between the separate parts. If records not fulfill the requirements of reliability and authenticity, they can not be used as evidence and not enable accountability (Thomassen, 2001).

Though many problems still remain to be solved, governmental as well as business organizations are in an intermediate stage in the changeover to electronic records and have to deal with this situation. In this paper an empirical study of seven Swedish governmental departments is presented.

2 The Study

The aim of the study was to investigate how different governmental departments managed electronic information both in a short and long run. As the organizations today handle both paper documents and electronic information, an overarching question is to see in which stage of changeover the organizations are in and which strategies they use to handle the mixture of paper document, scanned documents, electronic records, databases, etc.

2.1 Study object

The seven Swedish governmental departments that participated in the study were:

CSN

The national authority that handle the Swedish financial aid for students; i.e. loans and grants for studies.

The Swedish tax authority

Administration authority for all kinds of taxation, national registration and estate inventories

The Swedish Companies Registration Office (SCRO)

SCRO is a new government agency under the Ministry of Industry. Their operations are not financed by public funding and they therefore charge for the services they provide.

The Swedish Enforcement Administration and legislation

The authorities of the Enforcement Administration are responsible for the enforcement of both public and private claims. Public claims are debts to central and local authorities (taxes, VAT, excise duties, social security contributions but also for instance television licenses and parking fines). Private claims are based on judgments by general and administrative courts.

The National Government Employee Pensions Board (SPV)

Pension administration involves applying the rules of pension agreements and computing and paying out the different components of the pension.

The Swedish Social Insurance Administration

A new integrated government agency with responsibility for the Swedish social insurance system

County Council of Jämtland

The county council is responsible for several issues overarching the county, such as public transportation, culture and development of the county. However, the main task for county councils is medical and health care.

2.2 Method

The study was carried through with help of interviews and in examination of complementary documentation. The interviews were built up to follow an "object" or a commission from the initiation to the end and/or the filing for long term preservation. All additional information during the "objects" term of life was identified and questions about how different versions of the record was inquired. The "objects/commissions that were examined were:

CSN – one individual's loan from initiation to final repayment or removal from the cause list.

The Swedish tax authority – one individual's declaration for tax return to final assessment for income tax

The Swedish Companies Registration Office (SCRO) – one company from the start (though a company may come to an end it must not be sorted out)

The Swedish Enforcement Administration and legislation – one claim from the application to execution

The National Government Employee Pensions Board (SPV) – one individual's pension from saving to the person's death

The Swedish Social Insurance Administration – one individual's sick report from reporting to closing (fit declaration, pension, or death)

County Council of Jämtland – one individual's case record from birth or moving in.

2.3 Result

All of the studied objects were quite complicated and consisted of several steps and iterations. To exemplify a simplified picture of the information exchange for an individual's loan is shown in figure 1. Every step can be iterated several times. The example shows only a loan that is repaid in due time. What is not shown is if a person applies for respite for payment or refuses to pay, which causes separate decision processes and involves other authorities.

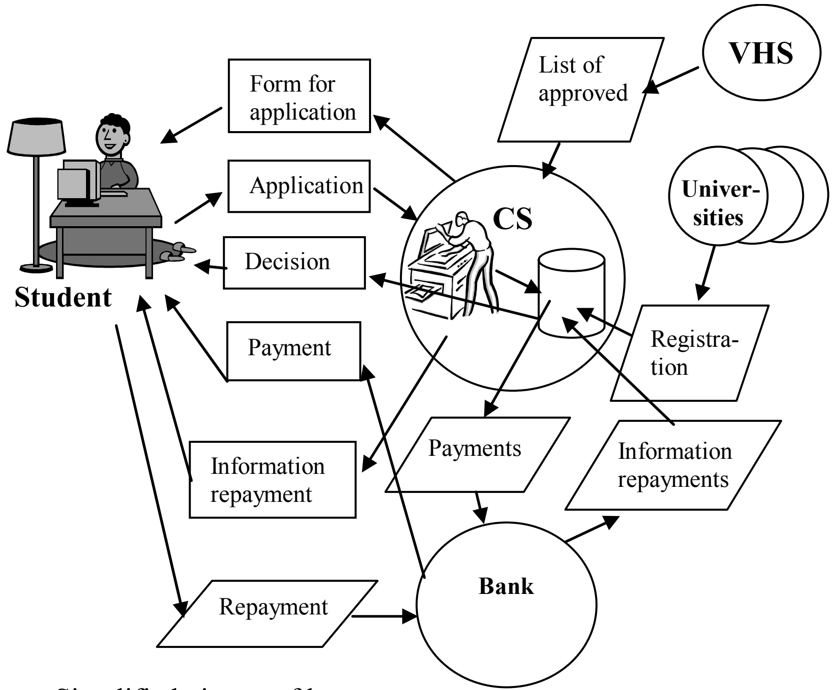


Fig. 1. Simplified picture of loans

The ambition of many governmental departments is to become 24/7 agencies. In table 1 can be seen how far the studied organizations have reached. Of the studied organization, four have e-services for some step in the process, though with varying shares of use. Two others plan to start e-services. The county council will not, for obvious reasons, supply any e-service for case records. All the studied organizations handle electronic records in the studied cases. They saved the electronic records in XML or other formats that is said to be appropriate for long-term preservation. Most of them also scanned paper documents and used them as electronic information. Much information is also stored as database records.

Table 1. E-service providing

Organization	E-service	E-records
CSN	Yes	Yes
Tax authority	Yes	Yes
SCRO	Yes, limited use	Yes
Enforcement Administration and legislation	Yes	Yes
SPV	No, planned	Yes
Social Insurance Administration	No, planned	Yes
County Council	No	Yes

The active times for the studied objects vary quite a lot, but mostly the information is active for decades (see table 2).

In some of the organizations is not all information preserved during that time, some sorting out is performed. Information that is sorted out is for example notes, e-mails, etc. As the amount of information is huge, just a selection is obliged to be preserved. However several of the organizations chose to save more than they were obliged to. Many of the organizations chose to save all the database records and did not sort out any of them. In some cases scanned documents were chosen to be the original and thus preserved. Other chose to print out the information and preserve paper documents. It deserves to be mentioned that electronic records has not existed for so long time and a feeling of uncertainty about how to do was expressed.

Table 2. Preservation for short and long term

Organization	Max active time	Long-term Preservation
CSN	Lifelong	A selection is preserved forever
Tax authority	5 years after declaration	A selection is preserved forever, in some special issues are all preserved
SCRO	"For ever"	Yes
Enforcement Administration and legislation	20 years	20 years
SPV	Lifelong	Yes
Social Insurance Administration	Several years	A selection is preserved forever
County Council	Lifelong	Yes

As information about an object or a commission is present for a long time an interesting question is whether different versions existed and was preserved. During the active time of the object several different documents were gathered. All documents were time stamped and thus it is possible to reconstruct different stages in the process. There are no assembled versions at different times. The information is rather accumulated over time. In the databases the records are dated, but there is far from any likeness of temporal databases. To be able to get a complete version of the information at a certain time an extensive selection and reconstruction of information is needed. Unlike paper-bound documents bits and pieces of information has to be put together to form a complete electronic record. As some information is paperbound and other is electronic a referential system is needed. Some organizations chose to have the references in a computer-based system. Other chose to create a folder with paperbound documents with a list of references to the electronic information.

Stove-piped organizations and systems, which have been predominant for a long time and still is very common in governmental organizations is showing weaknesses (O'Looney, 2002; Friday, 2003; Thakur & Satter, 1998; and Sundberg & Wallin, 2005). To manage more far-reaching commissions a stove-piped organization will be a hindrance. Some kind of business process reengineering would be preferable. Though the studied organizations are of a stove-piped character, they manage to act in a process-oriented way when it comes to manage the information about commissions.

3 Conclusion

The result of the study shows that all studied organizations are in some stage of changeover from paperbound information to electronic records. There is a lack of knowledge about how to handle the situation, but they have all developed different strategies to handle the mixture of paper documents, scanned documents, electronic records, and databases.

An increasing amount of electronic information is to be long-term preserved. There is very little knowledge about how this should be done. Guidelines must be developed and made available to those responsible for the preservation.

There is very little support in the systems for management of distributed information, information that is not only distributed in where it is stored, but also in what type of information it is. New tools, able to assemble the information in a transparent way, have to be developed.

References

Asproth. V. (2005) Information Technology Challenges for the Long-term Preservation of Electronic Information. International Journal for Public Information Systems, no 1, pp. 27-37.

Borglund, E. (2005).

Chen, Su-Shing (2001) The Paradox of Digital preservation, IEEE Computer, 34(3):24-28.

Duranti, L. (2001a) Concepts, Principles, and Methods for the Management of Electronic Records, The Information Society, vol. 17, pp. 271-279.

Duranti, L. (2001b) The impact of digital technology on archival science, Archival Science, vol. 1, pp. 39-55.
Friday, S. (2003) Organization Development for Facility Managers: Tracing the DNA of an FM Organization, Amacom Books.
Goldstein, H. (2004) The Infinite Archive, IEEE Spectrum, Vol. 41, p 74.
Grimson, J. (2001) Delivering the electronic healthcare record for the 21st century, International Journal of Medical Informatics, vol. 64, pp. 111-127.
Ketelaar, E. (1999) Archivalisation and archiving, in: Archives and manuscripts 27, pp. 54-61.
O'Looney, J. A. (2002). Wiring Governments. Challenges and Possibilities for Public Managers. Westport, CT: Quorum Books.
Statskontoret (2003) "Förstudierapport om framtidens lektroniska arkiv," 2003/67, 2003.
Sundberg, H., Wallin, P. (2005) Co-ordination of business and IT development processes – managing stovepiped organizations. International Journal for Public Information Systems, 2005:1, pp. 53-69.
Thomassen, T. (2001) A First Introduction to Archival Science, Archival Science, vol. 1, pp. 373-385.

The Managing and Complex Querying of the Digital Medical Images Collections

Liana Stanescu, Dumitru Dan Burdescu, Marius Brezovan

University of Craiova, Faculty of Automation, Computers and Electronics, stanescu_liana@software.ucv.ro, burdescu@topedge.com, brezovan_marius@software.ucv.ro

Abstract. The article presents a software tool working on-line with a multi-threaded client/server architecture that permits loading the images and alphanumeric data in a database either directly, or by processing a DICOM file. It also allows the simple text based query of the database and content-based query on color or color texture features at the level of the entire image or at the level of the regions. The software tool has a modularized architecture. The module for extracting the images and alphanumeric information from the DICOM files and the module for extracting image visual characteristics (color and color texture) and the color regions are on the server side and are implemented using Java technology. For the database the MySQL server is used, and the client interface which allows the execution of simple text based query and content-based visual query on color and color texture features is implemented using the JSP and Servlet technologies.

1 Introduction

With the development of the different technologies, in the medical field an impressive number of gray and color images is accumulated. A big part of these images comes from the medical devices satisfying the DICOM standard, providing a file with a standardized format that includes a multitude of alphanumeric information (the patient's name, the diagnosis, the time and date of the consult, the doctor's name, etc), and also one or several gray or color images, which can be compressed or not [4]. Other images come from digitizing the images stocked on different media (X-ray film, paper, etc). In the case of the existence of this large quantity of imagistic and alphanumeric data, a computerized management and their structuring in a database are necessary. The reasons are:

1. The DICOM file can't be visualized directly on the computer, it has to undergo a processing operation in order to extract the necessary information and to store in the database;
2. Creation of a database at the hospital level, its development at the regional and even national level in a distributed manner, allowing to keep track in time of the patient evolution; for example, a query on patient name can be launched, and all the necessary data (diagnosis, images) for each recorded consult is retrieved;
3. The database can be queried based on the image content and can be used in the processes of diagnosis, education and medical research;

The content-based visual query can take into consideration the entire image or only regions and this means establishing an image or a region as query in order to retrieve the images or regions from the database that are similar with the query [Smith JR 1997; Del Dimbo A 2001]. From the conversation with the doctors, the next situation appears frequently: the doctor visualizes a medical image, doesn't know exactly the diagnosis, but he is aware of the fact that he has seen something similar. The problem is that he doesn't have any possibility to search for something similar in the database. This problem can be solved setting that image as a query image. The content-based image query will provide the similar images from the database and it is very likely that among the retrieved images should be the searched image together with its diagnosis, observations, treatment. So the content-based visual query can be directly used in the diagnosis process. The visual characteristics (color, texture, shape) allow not only the retrieving of the patients having the same disease, but also the cases where the visual similitude exists, but the diagnosis differs, that can be useful in the medical research or medical education process.

There are few systems that are already integrated into the medical diagnosis process and the work for the application of the most suitable algorithms in image processing and features extraction continues [Muller H, Michoux N, Bandon D, Geissbuhler A 2004].

The most common type used for this kind of architecture is the client/server model [Djeraba C, Sebe N, Chang E 2006]. In this paper the multi-threaded client/server architecture of a software tool is presented. It permits the loading in a database of the images and alphanumeric data directly or by processing of the DICOM files and the querying of the database based on text or based on the image content represented by color or color texture features. The software system can run on different platforms (Unix, Windows, Linux), is developed using Java technologies, being robust, portable and extensible.

2 The Software Tool Architecture

In the case of the content-based visual query systems, it is desirable that the client interface, the database management system and the image processing module be executed on different computers. In the present case, using the JSP and Servlet technologies, the client will be able to access the service from anywhere using a Web browser and having an Internet connection. The database and the modules for DICOM files and image processing are executed on more powerful servers.

The general architecture of the software tools with modules and their connections appears in figure 1.

The software tool is designed using the client-server architecture. The module which processes the DICOM files by extracting the images and the alphanumeric data and the module which processes the images by extracting color, color textures features and color regions are on the server side and implemented in Java. The client graphical interface is implemented using Servlet and JSP technologies. Another module of the tool is the database server implemented using MySQL. The communication with the server is done by means of JDBC.

The communication protocol between the server and the client is a simple one:

- When DICOM file is uploaded, it is added first in the database (the client communicates directly with the database server by means of JDBC); the file is saved on the web server;
- The client sends to the server the unique ID of the file from the database; the server reads the file being processed, the alphanumeric data and the image(s) being extracted; the visual characteristics of the image are also extracted; the necessary information is saved in the database (DICOM file and image processing server and the database communicate by means of JDBC);
- The query requests are initiated by the clients and sent to the database server;

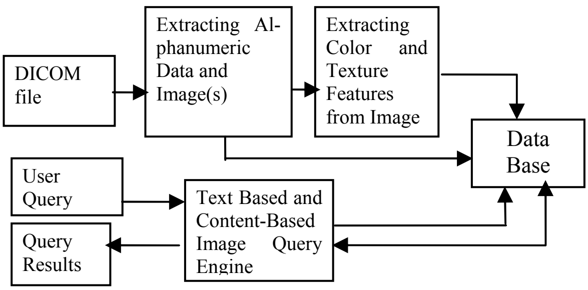


Fig. 1. The modules of the software tool

3 DICOM File Processing Server

It is the one of the most important module of the software tool and realizes the following functions:

- The processing of the DICOM standard files and extracting the alphanumeric data from tags and the image(s)
- Storing the generated data in the database

A DICOM file has the following structure [DICOM Homepage 2006]:

- A preamble of 128 bytes
- Prefix (4 bytes) where are stored the letters ‘D’, ‘I’, ‘C’, ‘M’ which represent the signature of the DICOM file
- Data Set that stores a set of information, such as: patient name, type of image, size of the image, etc.
- Pixels that compose the image (s) included into the DICOM file.

Extracting the data from the DICOM file can be made by taking into account every tag from the DICOM dictionary. This will be searched in the file and in case of finding it, the corresponding value will be extracted.

The steps of extracting information from DICOM files are:

- Verifying the existence of the ‘D’,’I’,’C’,’M’ characters
- Establishing the type of VR (ExplicitVR or ImplicitVR). This information is given by the UID (Unique Identifier), information stored in value field corresponding to the Transfer Syntax Tag.
- Establishing the Byte Ordering (BigEndian or LittleEndian). The information is also given by UID, stored in value field of the same Transfer Syntax Tag. The DICOM standard contains all the values that UID can have.
- Searching a tag in DICOM file according to the VR type and ByteOrdering
- Value extraction of the corresponding found tag.

Next it is described the problem of image extracting from the standard DICOM files. The images from DICOM files can be classified using several criteria [DICOM Homepage 2006]:

- The number of images stored in a file: single frame or multi frame.
- The number of bits per pixel: 8 bits, 12 bits, 16 bits or 24 bits.
- The Compression: without compression (raw) or with compression (RLE or JPEG).
- Photometric interpretation: gray scale images, color images, palette color images.

In the case of the images without compression, the extraction of pictures is made pixel by pixel, taking into account the number of bits stored for each pixel and the photometric interpretation (for monochrome images a pixel is stored using maximum 2 bytes and for color images, a pixel is stored using 3 bytes). In the images that use compression it is necessary a decompression algorithm before saving.

The pseudo code for retrieving the frames is:

```
Set variable number to 0
Loop until all frames are retrieved
        Set file dimension as Rows*Columns* (SamplePerPixel)
        Read all file dimension pixels from file starting with (Header Length +
        number* file dimension)
```

```
    If MONOCHROME image
        Save as image
            Store image using GIF or JPEG image format
        Return
    End If
    If PALETTE COLOR image
            Get the Palette tables (one for red values, one for green values
            and one for blue values)
        Get Pixels color from the Palette.
            Save as image
            Store image using GIF or JPEG image format
        End If
    If RGB image (24 bits)
            Get red, green and blue values
            Compute the color using the formula:
        ((255<<24) | ((0xff&r) << 16) |
        ((0xff&g)<< 8) | (0xff&b))
            Save as image
            Store image using GIF or JPEG image format
      End If
End Loop
```

4 Image Processing Server

It is another important module of the software tool and realizes the following functions:

- Extracting the visual characteristics (color and color textures) from images
- Detecting the color regions from images
- Storing the generated data in the database

Content extraction, indexing and retrieval of image and in general multimedia data continue to be one of the most challenging and fastest growing research domains. Primitive features (object motion, color, texture, shape) are generally extracted automatically and computed efficiently [Djeraba et al 2005; Hanjalic A, Sebe N, Chang E 2006; Lew M, Sebe N, Djeraba C, Jain R 2006].

The color is the visual feature immediately perceived on an image [Del Dimbo A 2001; Sebe N, Lew MS 2000; Smith JR 1997]. The color histograms represent the traditional method of describing the color properties of the images. They have the advantages of easy computation and up to certain point are insensitive to camera rotating, zooming, and changes in image resolution. In content-based visual query on color feature is important the used color space and the level of quantization, meaning the maximum number of colors. This software tool uses the representation of images in the HSV color space that has the properties of being complete, compact, natural and uniform and its quantization at 166 colors [Smith JR 1997]. The quantization algorithm generates a characteristics vector of maximum

166 values (that is stored in the database) and has the complexity O(width*height) where width and height represent the image dimensions [Burdescu D 1998].

Together with color, texture is a powerful characteristic of an image, present in nature and medical images, where a disease can be indicated by changes in the color and texture of a tissue. A series of methods have been studied to extract texture feature [Del Dimbo A 2001]. Among the most representatives methods of texture detection are the co-occurrence matrices and Gabor representations, implemented in this software tool [Del Dimbo A 2001; Palm C, Keysers D, Lehmann T, Spitzer K 2000]. There are many techniques used for texture extraction, but there isn't a certain method that can be considered the most appropriate, this depending on the application and the type of images taken into account.

In the case of the method based on co-occurrence matrices for the color images, one matrix was computed for each of the three channels R, G, B.

For an image f(x, y), the co-occurrence matrix $h_{d\phi}$ (i, j) is defined so that each entry (i, j) is equal to the number of times for that:

$f(x_1,y_1) = i$ and $f(x_2,y_2) = j$, where $(x_2,y_2) = (x_1,y_1) + (d\cos\phi, d\sin\phi)$ [Del Dimbo A 2001].

This leads to three quadratic matrices of dimension equal to the number of the color levels presented in an image (256 in this case) for each distance d and orientation ϕ. The classification of texture is based on the characteristics extracted from the co-occurrence matrix: energy, entropy, maximum probability, contrast, inverse difference moment and correlation [Del Dimbo A 2001]. The three vectors of texture characteristics extracted from the three co-occurrence matrices are created using the 6 characteristics computed for d=1 and ϕ=0. The texture representation in this case is done using a characteristics vector with 18 values stored in the database.

The functions that implement this method have the temporal complexity $O(m^2)$, where m is the maximum dimension of the image [Burdescu D 1998].

The computation of the Gabor characteristics for the image represented in the HS-complex space is similar to the one for the monochromatic Gabor characteristics, because the combination of color channels is done before filtering [Palm C et al 2000]:

$$C_{f,\varphi} = (\sum_{x,y} (FFT^{-1}\{P(u,v) \cdot M_{f,\varphi}(u,v)\}))^2 \tag{1}$$

The Gabor characteristics vector is created using the value $C_{f,\varphi}$ computed for 3 scales and 4 orientations [Palm C et al 2000]:

$$f = (C_{0,0}, C_{0,1}, \ldots, C_{2,3}) \tag{2}$$

So, the texture representation in this case is done using a characteristics vector with 12 values stored in the database.

The implemented Gabor filter procedure has a temporal complexity $O(n^2)$ where n is the maximum dimension of the image [Burdescu D 1998].

For detecting color regions it was chosen the color set back-projection algorithm, introduced initially by Swain and Ballard and then developed in the research projects at the Columbia University [Smith JR 1997]. The extraction system for color regions has four steps [Smith JR 1997]:

1. the image transformation, quantization and filtering (the transformation from the RGB color space to HSV color space and the quantization of the HSV color space at 166 colors; the binary color set with 166 values is computed-value 1 representing the present color and value 0 representing the missing color)
2. back-projection of binary color sets
3. the labeling of regions
4. the extraction of the region features

For each detected regions the color set that generated it, the area and the localization are stored. All the information is necessary further on in studying the evolution of the patients. The region localization is given by the minimal bounding rectangle (MBR). The region area is represented by the number of color pixels and can be smaller than the minimum bounding rectangle. Two original implementations of the color set back-projection algorithm are presented in [Burdescu D and Stanescu L 2005].

The application of an automated algorithm for detecting the color regions in medical images has two important utilizations:

1. in content-based region query on medical images collections, the specialist chooses one or more of the detected regions for querying the database, the purpose being the retrieval of images similar by color and texture; this can be useful for clarifying some uncertain diagnosis. This problem was studied in [Stanescu L and Burdescu D 2003] interesting results being presented.
2. during the evolution in time of the disease of patients that follow a certain treatment

5 The Database Server

The Entity-Relationship model for the structure of the database used by the software tool is presented in figure 2. The resulting tables are the following:

Tags table contains the following information:

1. tag – all the tags from DICOM Dictionary; for example: (0010, 0010);
2. the name of each tag; for example: Patient Name;
3. the Value Representation of each tag;
4. the Value Multiplicity of each tag;
5. the Version;

6. the Category of each tag; for example the information about the patient goes into Patient Category

Dicom_Files table contains:

- the unique identifier of the file;
- the path to that file;

Images table contains:

- the unique identifier of each image stored;
- the unique identifier of the DICOM file from where the image was extracted;
- the number of the frame (in case of DICOM file with multiple frames);
- the path and the name of the image;
- the color characteristics vector;
- the texture characteristics vector

Header table contains the values of the tags retrieved from DICOM files:

- the unique identifier of each entry in the table;
- the unique identifier of the DICOM file from where header information was retrieved;
- the tag;
- the extracted value;

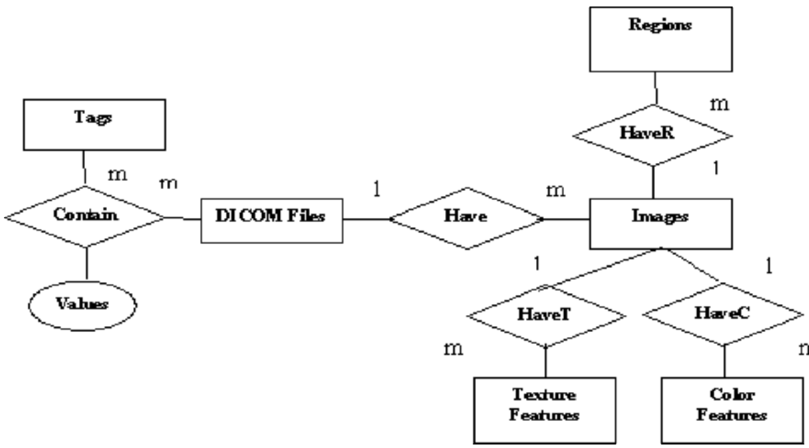


Fig. 2. The structure of the database

The Header table appears as a result of a m:m relationship between the tables Tags and Dicom_Files.

Regions table contains:

- the unique identifier of the detected region;
- colorset (the color set of the region);
- the coordinates of the upper left corner of the minimum bounding rectangle;
- the coordinates of the bottom right corner of the minimum bounding rectangle;
- the effective number of pixels corresponding to the considered color set, that is the surface of the region

6 The Client

The graphical interface plays an important role in the retrieving process. It must be, as much as possible, intuitively and friendly, permitting to the users that are not experts in the field, to navigate and search in the image collections. They don't need knowledge about image processing, about the mechanism of extracting visual characteristics, or about the measuring the images dissimilitude.

Also, the interface must permit the experts to improve the content-based query process using advanced settings of the parameters or combining the visual characteristics.

In the presented software tool, the client can issue four types of queries, namely:

- simple text-based queries;
- image queries based on color feature;
- image queries based on texture;
- region queries;

Some of the client functions are the following:

- The user can browse through the images;
- The user can design its own case by uploading DICOM files or images directly; these are processed and stored in the database;
- The user can search for images that are similar with the selected one on color or/and color texture features; also, the user can search for one or more regions similar with the query region taking into consideration sizes like color, area, absolute position. The user can set different values for the appropriate parameters.

For computing the distance between the color histograms of the query image and the target image, there have been taken into consideration the intersection of histograms, square distance between histograms and Euclidian distance [Smith JR 1997].

The reason why these three modalities for computing the images similitude were chosen is explained next. In the case of the medical images, the effectuated studies shown that the results were weaker than in the case of the nature images [Stanescu L and Burdescu D 2003]. None of the distances mentioned above caused much better results than the others, so that to motivate its choosing. In each case, relevant images for the query were retrieved with one of the distances and they were not retrieved by the others. It could be observed that in most cases, all the results retrieved by computing the three distances may be useful for not loosing relevant images for the query, consequently they complement one another.

In the case of the method based on co-occurrence matrices, the texture similitude between the query image Q and target image T is computed by the Euclidian metric.

In the case of the method based on Gabor filter, the similitude between the texture characteristics of the query image Q and the target image T is defined by the metric [Palm C et al 2000]:

$$\mathrm{D}^2(\mathrm{Q,T}) = \sum_f \sum_\varphi d_{f\varphi}(Q,T), \textit{where } \mathrm{d}_{f\varphi} = (f^Q - f^T)^2 \tag{3}$$

In the case of extracting color texture features two methods were also implemented, following the same above idea – the two resulting sets of images complement each other.

In figure 3 appears an example of user interface.

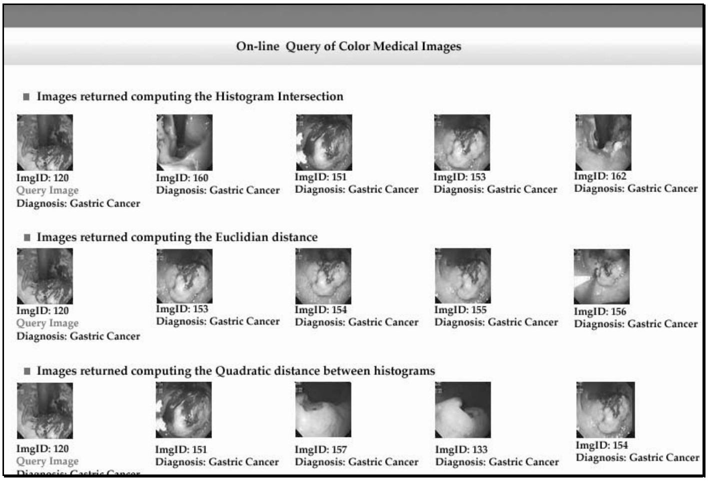


Fig. 3. User Interface: The results of the content-based visual query on color feature (three sets of results were computed)

7 Conclusion

The paper presents a Java based software tool with a multi-threaded client/server architecture and the following functions:

- The processing of the DICOM standard files provided by the medical tools in order to extract the alphanumeric data and images
- Storing the resulting data in a database
- Extracting the color feature from the image (the color histogram and the binary color set resulting from the HSV color space quantized at 166 colors)
- Extracting the color texture feature from the color images (using the co-occurrence matrices and the Gabor filters)
- detecting color regions and storing the new data in the database
- the simple text-based query of the database
- the content-based visual query of the database on color and color texture features
- the content-based region query of the database

The software tool is tested now by the medical personnel taking into consideration aspects like execution speed, retrieval quality and new necessary options.

References

Burdescu D (1998) Analiza complexitatii algoritmilor. Albastra, Cluj-Napoca

Burdescu D, Stanescu L, (2005) A New Algorithm For Content-Based Region Query in Multimedia Database. In: Lecture Notes in Computer Science 3588. Springer Verlag pp 124-133

Del Bimbo A (2001) Visual Information Retrieval. Morgan Kaufmann Publishers, San Francisco

DICOM Homepage (2006) http://medical.nema.org/

Djeraba C, Sebe N, Lew MS (2005) Systems and Architectures for Multimedia Information Retrieval. ACM Multimedia Systems Journal. 10(6) pp 457-463

Hanjalic A, Sebe N, Chang E (2006) Multimedia Content Analysis, Management, and Retrieval: Trends and Challenges. In: Proc. Electronic Imaging, SPIE'06, San Jose. USA

Lew M, Sebe N, Djeraba C, Jain R (2006) Content-based Multimedia Information Retrieval: State-of-the-art and Challenges. ACM Transactions on Multimedia Computing, Communication, and Applications. 2(1) pp 1-19

Muller H, Michoux N, Bandon D, Geissbuhler A (2004) A Review of Content_based Image Retrieval Systems in Medical Application – Clinical Benefits and Future Directions. Int J Med Inform. 73(1) pp 1-23

Palm C, Keysers D, Lehmann T, Spitzer K (2000) Gabor Filtering of Complex Hue/Saturation Images For Color Texture Classification. In: Proc. 5th Joint Conference on Onformation Science (JCIS2000) 2. Atlantic City, USA pp 45-49

Sebe N, Lew MS (2001) Texture Features for Content-based Retrieval. Principles of Visual Information Retrieval, Springer, ISBN 1-852333-381-2, pp 51-86

Sebe N, Lew MS (2000) Color Based Retrieval and Recognition. In: Proc. IEEE International Conference on Multimedia and Expo. New York. pp 311-314

Smith JR (1997) Integrated Spatial and Feature Image Systems: Retrieval, Compression and Analysis. Ph.D. thesis, Columbia University

Stanescu L, Burdescu D (2003) IMTEST-Software System For The Content-based Visual Retrieval Study. In: 14th International Conference On Control Systems And Computer Science. Bucuresti

Semantic Execution of BPEL Processes

Péter Martinek, Béla Szikora

Budapest University of Technology and Economics, Goldmann tér 3, Budapest, HU-1111

Abstract: Service Oriented Architecture (SOA) is an up-to-date software architecture, which is used in many areas of informatics today. There are lots of open standards and free tools which aim to realize the concepts of SOA.

This paper introduces an approach by using a given collection of open standards and tools to realize a semantically enriched SOA architecture. The main building blocks of the architecture, the Web services, will be extended by a semantic description. The resulting SOA architecture and the services themselves will be called semantically enriched. For orchestration we use the OWL-S [1] standard. The choreography will be described in BPEL. The semantically enriched Web services run on an OWL-S Virtual Machine, while the complex processes, described in BPEL [2] on a BPEL run-time. Between the OWL-S invoke and BPEL description a special, self-developed mediator service will be used.

In this paper we focus on the execution environment and the prerequisites to successfully build it, the publication of the service descriptions and discovery of the services will be only briefly introduced.

1 Introduction

Management of business processes is a key factor to enterprises today. Most of the business processes run through several different areas of information system functions, however the recently used business application systems (i.e: ERP, CRM, SCM, etc.) are rather isolated. Identifying the roles of partners and their functions on the processes can result in a process based integration scenario. These processes are mostly described by Business Process Execution Language (BPEL) which is a suitable open standard language to describe complex processes. The most important building elements of a complex process are as follows: **invoke** of services from information system, **assign** between variables come from the replies of the invoked services or constant values given by the designer itself and the **reply** of the results at the process endpoint. Although BPEL is a widely-used standard which is supported by reliable tools (designer with user-friendly graphical interface and run-time engines), designing a process is not an easy task to do. The builder of a process must know the specific services (more precisely: the location of the service descriptions) which will be invoked in the process run-time. Service registries can help to find the services by selecting the given categories (business entity, keywords, etc.) but searching according to the requested capability of the service is not properly solved yet. Enriching the services description with semantics and publish them in a searchable registry can solve the problem, as many paper has already been suggested. The attached semantic description carries information about service capabilities the input, output data of the service refers to a shared collection of concepts called ontology. This paper introduces an approach, which makes it possible to use existing BPEL tools combining them with the usage of OWL-S services and OWL-S run-time. We also show a small example which helps us to validate our approach.

2 Literature review

There are many research groups and publications which deal with (semantically enriched) SOA architecture today.

Large Scale Distributed Information System (LSDIS) group [3] at the University of Georgia has a SOA approach using BPEL for choreography as well, but the semantic descriptions of the Web services are in WSDL-S.

The Intelligent Software Agents Lab [4] at Carnegie Mellon deals with an SOA approach which is using OWL-S service description for choreography. They also developed tools for supporting the creation of the OWL-S description from a WSDL [7] basis and publicating them into a UDDI [8] registry. They, however, use the OWL-S to describe the processes as well, while we use BPEL for this purpose.

There are many tools and solutions to design and run standard BPEL processes, for example the Oralce Fusion Middleware [5] or the IBM Websphere [6]. But they don’t provide the extension to describe the services in OWL-S and discovery them by their capabilities.

3 Adding semantics to services descriptions

The most widely-used standard to describe a Web service is the Web Service Description Language (WSDL). A WSDL description contains every necessary information to invoke the service, but it tells very little about how the service works. The extension of the description with this information happens in the so called annotation phase. The annotation starts with migrating the information found in the WSDL into an OWL-S skeleton. After that, the skeleton will be extended with the following: functional capabilities of the service will be described in the **profile**

part and the input and output data referring to the concepts in our ontology take place in the **grounding**.

Because the services come from different vendors and are described only with a WSDL, the input and output data of these native services must be transformed into and from the concepts of the ontology. By invoking a service, the inputs written according to the concepts of the ontology (in the OWL-S description) must be transformed into the input data type according the type of the input message contained in the WSDL. This is called **downcast transformation** because the direction of the transformation is from an abstract higher level (level of ontology concepts) towards a lower one (level of (complex) XSD [10] data-types. By catching the output of the invoked service usually we get a (complex) XSD data-type, which must be transformed into the concepts of the ontology. This is called **upcast transformation**. The transformations can be written for example in XSLT [11] and can also be stored in the grounding part of the OWL-S description. (There are many languages to describe data transformation, but XSLT have been chosen, because the selected OWL-S VM directly supports the usage of XSLT. See more in section 2.)

The OWL-S description also allow to put services into a given structure (sequence, junction, etc.) creating complex services this way. This can be done in the **process part** of the OWL-S description. In our approach we use BPEL engine for choreography purposes, because we believe, that BPEL engines are more reliable and robust for this task.

4 Invoking semantic services

There are existing tools to invoke semantic OWL-S services. In our approach we use the OWL-S Virtual Machine (VM) and API from the Project Mindswap by the University of Maryland [9]. The java code including the invoke of the OWL-S services can be extended as a standard Web service. This Web service has already a standard WSDL description, so it can be built into a BPEL process using a standard BPEL editor. The input and output data of these mediator Web services are not concepts of the ontology, so that transformations between complex XSD types defined in this WSDL and the individuals from the OWL-S description (input and output of OWL-S service) must be put here and done by this service. (This is a different kind of transformation than the XSLT mentioned in section 1. in up- and downcast transformations.) Because this service is placed between the BPEL runtime and the OWL-S VM it is called **mediator service**. The mediator service to each OWL-S service must be done during (or right after) the annotation phase. Because the concepts of our ontology are in a well known structure the generation of the up- and downcast transformation part of the mediator service can be largely automated.

5 Using Existing BPEL editor

After the services were successfully annotated they are ready to deploy and published in a registry. By using a suitable designer the services can be found according the semantic information during the design of the complex process. While the discovery of the services happens according the OWL-S description, the building in is made according the WSDL of the given mediator service. (Most of the standard BPEL editors rely on the WSDL description of Web services during the design of processes.) During the creation of the choreography (the complex process) the assign of variables should be done as well. Because the BPEL editor works only with the WSDL description of the mediator services, data can only be manipulated using the types from WSDLs. According to this building the right structure (with well-defined recognizable names) from the OWL [12] individuals in the mediator service is essential. By creating an **assign** operation in the process, the concepts of the ontology can be easily identified this way in spite of that the designing is practically made on other data types.

The approach for the overall execution environment is shown in Figure 1.

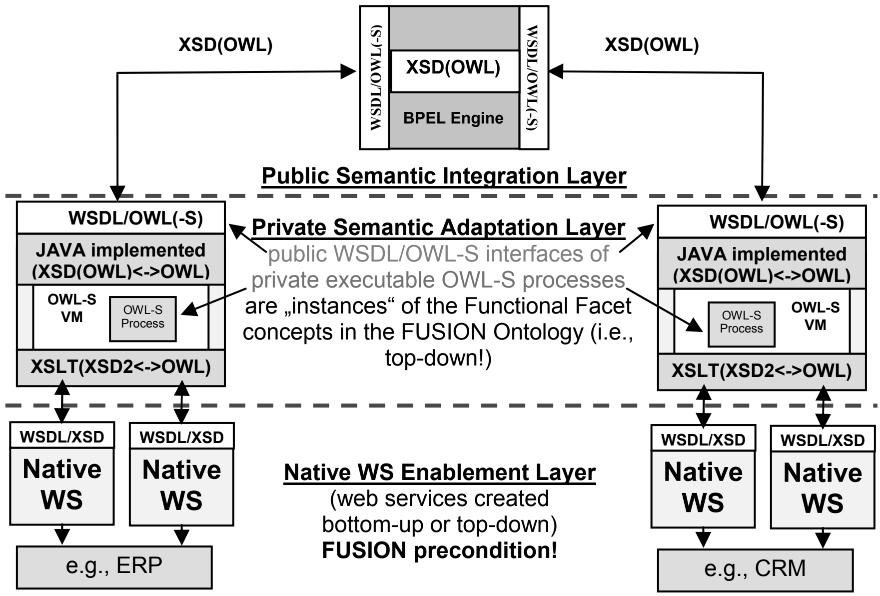


Fig. 1. The overall approach for the execution environment

6 Short Example

This section introduces short example which goal is the connection of two services. This means the realization of a process containing two Web services in a sequence. We consider implementing this process as a proof-of-concept that the approach is good enough to facilitate the specification of the execution environment and the overall solution. The Web services in the example were written in JAVA and published by the Eclipse environment as Web services. Testing of the Web services could be done easily by the Web service Explorer of Java, but other tools were used for testing as well, to make sure, that this solution (usage of Eclipse and Java implementation of Web services) fits to the current standards like SOAP, WSDL, etc.

The native WSDL services have been annotated according to our overall approach. The OWL-S descriptions were made by the Protégé OWL-S plugin. The descriptions of semantically annotated services contain the references to the standard WSDL grounding, plus the up- and downcast transformations between the native (WSDL) and the abstract ontology layer as well. To invoke these OWL-S services the Mindswap OWL-S API were used which is written in Java. It's input and output objects are standard Java objects but making automatically a Web service in Eclipse from them already gives us a native Web service with a standard Web service interface. The communication with the OWL-S services is done by OWL-individuals. By invoking an OWL-S service the input OWL individual must be sent to the OWL-S VM. By receiving the reply we get an OWL individual which we processed and transformed into a different Java object suitable for automatically transforming into a Web service reply. In our example these transformations were implemented hard-wired, which means, that two Web services were written for catching the Web service invoke from BPEL, doing these transformation, and invoke the OWL-S services. The overall BPEL process was designed according to the WSDL description of these Web services which acts as mediator between OWL-S and BPEL. In the BPEL we tried to use XML datatypes, which represents the concepts of the ontology (by using the same name, similar data-structure, etc.). In our simple example we needed to use only very simple assign operation (no XPath evaluations were needed, the input-output messages could be connected directly) between the two Web services, in a more sophisticated scenario we should use more complex assign operation, but it seems to be no problem from the point of our approach. The BPEL was designed and deployed by Active BPEL designer.

7 Summary

In the following points we summarize our results:

- Our approach deals with the proper annotation process of native (WSDL described) Web services. The OWL-S description contains references to the concepts of the Ontology for example the identification of the input and output of the Web service. The up- and downcast XSLT transformation code must be included in the same OWL-S description.

- OWL-S described annotated services are the building blocks of our processes, designed by the process designer.
- The invoking of the native Web services of the ERP and CRM systems is made by OWL-S Virtual Machine (we use the OWL-S API from the University of Maryland). This OWL-S API is responsible for running the up and downcast transformation as well.
- The execution of the process (which is an executable BPEL code) is done by the Active BPEL run-time engine. At the level of BPEL we use special XSD(OWL) data types (come from the WSDL of the mediator service) which are in a consistent connection with the concepts of the ontology.
- The mediator Web service (written in JAVA) is placed between the OWL-S VM and the BPEL run-time engine. This is responsible for catching the invokes from the BPEL, and transforming their input to the right OWL individual (according to the given OWL-S service which will be invoked) and catching the response from the OWL-S service and transforming them to the right XSD(OWL) type and forwarding it to the BPEL run-time.

The paper was supported by the **FUSION FP6-027385 Project**.

References

1. DARPA Agent Markup Language Working Group, “OWL-S: Semantic Markup for Web Services” at http://www.daml.org/services/owl-s/1.1/, 2004.11.
2. Business Process Execution Language for Web Services version 1.1“ at http://www-128.ibm.com/developerworks/library/specification/ws-bpel/ 2003.03.
3. Large Scale Distributed Information Systems (LSDIS) group, at http://lsdis.cs.uga.edu/ viewed 10.07.2006.
4. Intelligent Software Agents Lab, at http://www.cs.cmu.edu/~softagents/ viewed 10.07.2006.
5. Oracle Fusion Middleware, at http://www.oracle.com/technology/ products/middleware/index.html viewed 10.07.206.
6. IBM Websphere at http://www-306.ibm.com/software/websphere/ viewed 10.07.2006.
7. W3C, “Web Service Description Language, W3C Working Group Note” at http://www.w3.org/TR/wsdl, 2001.03.15.
8. OASIS, “UDDI, Advanced Web Services Discovery Standard” at http://www.uddi.org/specification.html, 2004.

9. Maryland Information and Network Dynamics Lab Semantic Web Agents Project, “OWL-S API”, at http://www.mindswap.org/2004/owl-s/api/ viewed 10.07.2006.
10. W3C, “XML and XML Schema”, at http://www.w3.org/XML/Schema viewed 10.07.2006.
11. [W3C, “XSL Transformation”, at http://www.w3.org/TR/xslt viewed 10.07.2006.
12. W3C, “Web Ontology Language (OWL)” at http://www.w3.org/2004/OWL/ viewed 10.07.2006.

MEO Ontology Infrastructure

István Szakadát*, Miklós Szőts**, György Gyepesi**

*Budapest University of Technology and Economics, i@syi.hu,
**Applied Logic Laboratory, Budapest, {szots, gyepesi}@all.hu,

Abstract: MEO project (Hungarian Unified Ontology) started in 2005. The most important goals of the project were 1) building an upper level ontology, 2) developing a special application supported by ontological knowledge, and 3) planning, establishing and distributing an ontology infrastructure. All achievements of the project were intended to be freely accessible and usable according to the Creative Commons philosophy. In the first part of the paper we outline the ontology building process, briefly describe the most important ontological concepts, present and interpret our most important methodological decisions and our ontological commitments, and show the applied layered solution of the project: how the conceptual and the language layers can be separated and integrated. While the conceptual layer is always language independent, the language layer can contain any number of individual languages without restriction. In the second part of the paper we discuss the structure and the logic of MEO ontology, and present some interesting problems we had to solve in our project.

1 Introduction

The MEO project (Magyar Egységes Ontológia/Hungarian Unified Ontology) started in 2005. Based on a wide academic and industrial partnership with seven consortium members the project was sponsored by NKFP (a Hungarian governmental R&D Program). The most important goals of the project were:

1. building an upper level ontology,
2. developing a special application supported by domain level ontological knowledge in the field of the telecommunication call center activities,
3. planning, establishing and distributing an ontology infrastructure, and
4. forming a framework for cooperation, consensus management during ontology building processes.

Although in recent years the category of ontology has functioned as a relatively new buzzword, we put the emphasis on our third goal. In spite of the continuously increasing popularity of ontologies, it seemed to be evident to us, that our most important, current task is to learn how we can handle our ontologies, rather than build ontologies that we do not know how and for what purpose we can use. Of course, by the end of the project we shall build a top level and a domain ontology, but it will be more important to have a tool set (a special integrated, consistent ontology infrastructure component set), with the help of which we or anybody else can start a new ontology building process.

While the knowledge we build into our ontologies consists of words, expressions, concepts which are the results of constant human cooperation over the history of mankind, we decided at the beginning of the project that all achievements of the project would be freely accessible and usable by the public (in line with the Creative Commons philosophy we retain only the Attribution licence to all important achievements of the project). The project communication was fitted to our Creative Commons based commitment, and we launched a MEO session on the Hungarian ontology portal (ontologia.hu/meo), where we made all our project achievements (official project reports, working papers, models, manuals, ontology components etc.) freely accessible.

Of course we did not want to reinvent the wheel. From the beginning of the project it was obvious for us, that we would have to try to reuse all freely available resources. After composing the first version of our upper ontology, we compared our top level concepts to elements of possibly reusable upper level ontologies (like SUMO, WordNet, Dolce). For example we mapped SUMO's time and location concepts to the ones in MEO. Sometimes we adopted good solutions, useful concepts from SUMO and Dolce. But it was the area of ontology building methodology, which had really great influence on our project. We learnt a lot especially from the OntoClean methodology. Although we started to develop our own lexicon and ontology editor (MEOditor), a very popular, widely used ontology editor (Protégé) was also included into our suggested ontology infrastructure tool set. Our editor, of course, supports the two most important ontology related formal languages (OWL and DL), and we provided export/import utilities between MEOditor and Protégé as well.

2 Layered approach

In the MEO project we adopted a layered solution in order to ensure the language neutrality of the ontology. The task is obvious. An ontology – per definitionem – is language independent, but if we (human contributors) would like to use ontological knowledge, we necessarily need a language that is bound to it (a natural or an artificial language). In our MEO model we distinguished a language and a concept layer. We can build our ontologies within the concept layer – totally independently of any language, and in the language layer we can connect as many language dependent words, expressions to our concepts, as we wish. Due to this separation we can build our ontologies in a truly language independent way, and we do not have any language limits. An ontology can be built in any number of languages, and it can not present a problem if a certain language lacks a linguistic construction for a particular concept.

In the MEO model not only the language and the conceptual layers were separated, but we made another distinction as well. In the first phase of the project we could deal with our concepts only on a general level which meant that we divided our concept layer into two parts: a generic level and an instance level. The generic part, the so-called concept domain contains concepts (not instances), and in the other part of the model, what we called instance domain, we can build – in the future – our instance level knowledge base.

Based on the two dimensions briefly introduced in the previous paragraphs we show in Figure 1. how the different layers of the MEO model can be separated and integrated.

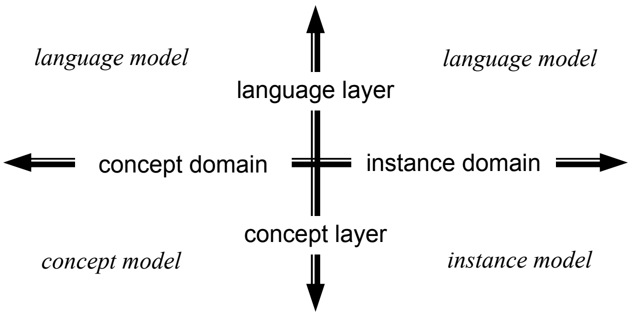


Fig. 1. Parts of MEO model

The language layer has its own model (a linguistic ontology). The elements of the language model provide a meta-ontology for our object level ontology. Although our model is valid at the levels of both spoken and written language, in our project we restricted our work to the written language domain.

The most important entities of the MEO linguistic ontology are:

- wordform
- morphological unit
- construction (similar to lexical unit)

Wordform is a simple string without any language binding. This is an entity where we need not differentiate the English word ‘nap’ (as an activity) from the ambiguous Hungarian word ‘nap’ (as a period (day) or a star (sun)). This level is needed because it is only here that we can assign word frequencies to the simple strings independently of the languages the wordforms may be found in. If we bind a language and a set of morphological features to a selected wordform, we can talk about a *morphological unit*. Based on this type of information we can predict the behavior of the wordform in sentences. But this grammatical information is not enough if we wish to know the sense of the selected wordform, so we need – on a third level – the entity of the *construction*. With a construction we can bind a cept (and its sense as well) and a wordform or morphological unit (a language specific utterance). Construction is very similar to the category of *lexical unit*, but the two notions are not identical. In our adopted linguistic theory lexical units are *construction constants*, but we need *construction functions* as well if we want to grasp all other types of linguistic phenomena (for example if we want to describe the productive derivation of words).

MEO has a strong commitment to construction grammar, which is not particularly widely used either in the ontology building community or in the field of linguistics. This is a unique feature of our project.

The next figure shows the most important linguistic relationships:

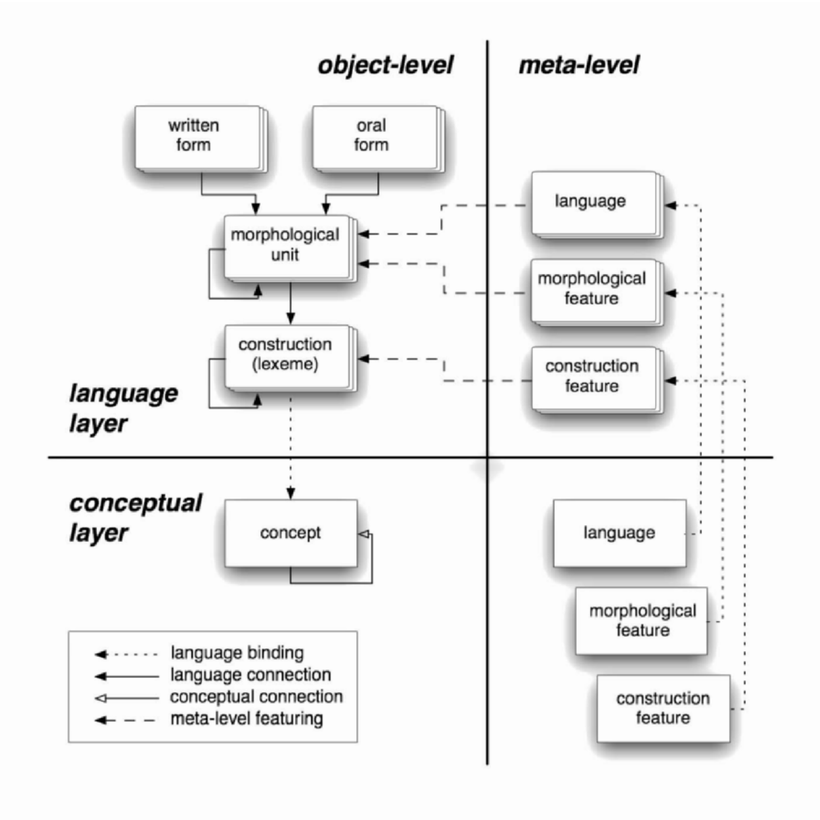


Fig. 2. Linguistic relationships of MEO

3 Concept model

The concept model is the most important part of the entire MEO model in the intersection of the concept layer and the concept domain. Within our model we needed to separate precisely the object level and the different meta-level concepts and concept areas. 1) At the most abstract level we have some mathematical concepts, of course, 2) we have an object level concept domain where we can build our ontological concepts, and 3) we have another meta-level area, where we separate and handle those meta-concepts that we need to describe our ontological concepts.

The function of the mathematical level having only three main categories is only to transform the concepts of ontology into the language of mathematical logic. We use the notion of *relationship* as is usual in mathematics, and we have two other subordinate categories, the notions of *first order relationship* and *higher order relationship* (with the usual interpretations).

Our basic meta-level entities are *meta-properties*, *meta-relations*, *operations* and *primitive types*. We declared and used the same primitive types that the programming world uses – these are roughly same the as the data types used in OWL. If it is necessary or simply useful, some meta-properties can be introduced into this meta-level of our model, such as rigid, semi-rigid or dependent in the OntoClean methodologies, but, of course, we can define any other types of meta-property. On this level we defined the most important relation properties (symmetric, asymmetric, transitive, intransitive etc.), and based on these properties and some consequences of them we can build and use a special relation property checking mechanism without using any complex inference system. On this meta-level of the MEO system meta-relations can be declared and used as well, such as the *generic* or *disjunction* relations (this term came from thesauri, and instead of 'generic relation' we could use a lot of other alternative terms like subclass, is-a-kind-of, AKO, subsumption etc.). The generic relation has a special and very important role in our model, we use it on various levels with the same interpretations, but between partly different entities (for example it is used between object-level concepts or between object-level concepts and mathematical entities). Finally, operations can be declared and implemented on this meta-level in order to create concepts from concepts with the help of operations on the object level. We introduced a lot of OWL and DL operators into the MEO model, such as union, intersection, complement, minimum or maximum cardinality, inverse, composition etc.

The main entity of the object level is the *concept*. The most important goal of the ontology building activity is creating new and new concepts, characterizing them with the help of different features, and establishing relations among the concepts. We differentiated three subtypes of concept, and declared *class concepts*, *relation concepts*, and *attributes*. The difference between class and relation concepts is obvious, we interpreted these categories as it is common in the database and programming world. Attribute is a relation-like concept, but attribute and relation differ from each other in their ranges. While the domain and the range of the relation concept can come from class concepts, in the case of attribute the range can come from primitive types. In this respect attribute overlaps the object level and meta-level area of our model. In our approach the three main subordinated concepts have totally equal status, and it is not so common in the ontology building community. Most ontologies concentrate on class concepts. Our ontology contains not only the concept of 'father' (which is a class concept), but the 'is father of' relation is considered as concept. In our universe there can be another relation type, a "*free*" *second order relation*. In this context 'free' means it can be freely declared and defined by the editors of the special ontology. A possible example for it can be the so-called evolutional development relation, with which we can describe the relationship between Horse and Przewalski's Horse in the taxonomy of animals. This relation is not generic one, because there is no connection between the real instances of the two concepts, and it can not be defined by a first order formula (in contrast to generic relation). It is a pure second order relation, since it can be interpreted between two concepts.

The different parts and the most important entities of the concept model are the following:

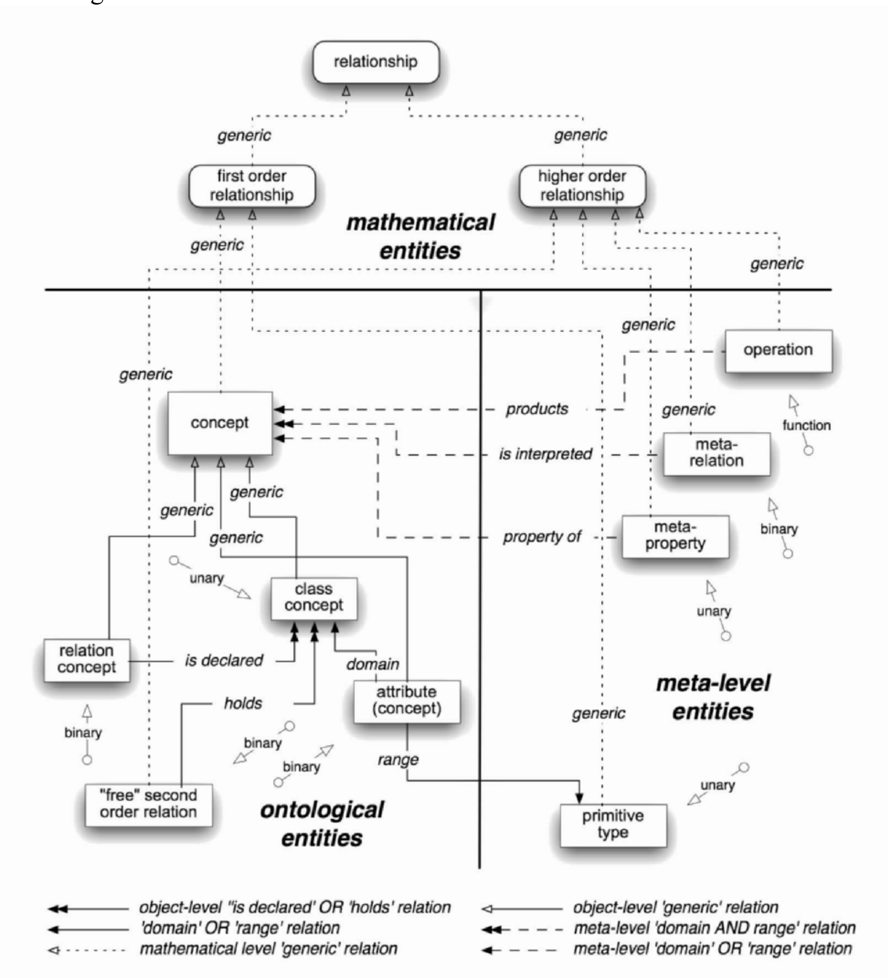


Fig. 3. Different parts and the most important entities of the concept

Under the highest level, most abstract conceptual entities, the model has some other concepts which can be used in all types of ontology. Very closely connected to the generic relation, we defined and applied the notions of *partition* and *hierarchy*. We declared a very important ontology building rule: in the object level area of the model, during the ontology building process we have to subordinate our new concepts to the already existing concepts in a given ontology. As we have three concepts subordinate to our main category (concept), all new concepts have to be subordinated to these (class, relation and attribute) concepts, and therefore potentially every ontology can have three disjunct hierarchies (generic relation based structures).

For each hierarchy we introduced a new concept type which we called *category*. Category (in our model) is a concept immediately subordinate to the root concept of the possible hierarchies. This notion (or more exactly this interpretation) came from the field of the classification systems and thesauri. In general ‘category’ is a widespread and overused term, but in our interpretation it has a definite, unambiguous meaning, and we can say, that categories are the first level concepts in the concept hierarchies.

4 Concepts and features

In the MEO project we tried to reuse a very old philosophical category first applied by Aristotle. When Aristotle described how we could create new concepts, how we could build taxonomies, he used the notion of *differentia specifica*. When we create a new concept with the help of the generic relationship we have to provide a feature, which unambiguously characterizes the new concept. This idea is very simple and seems to be very promising, although the well-known ontologies (or thesauri) did not adopt it. The probable explanation is again very simple. Providing new features every time we create new concepts would require much more effort. Based on this principle we should build a dual taxonomy, because we have to provide the same number of new features (differentia specifica) as the number of concepts we have. It is not an easy task.

In our project we did not require differentia specifica to be assigned to every new concept, but we incorporated this “featuring” possibility into our model. We can characterize our concepts in different ways. The attributes and the relation concepts can be described with the help of the well-known relation algebraic properties, such as symmetric, irreflexive, transitive etc. In the case of class concepts we can use meta-properties (the kind, for example, that OntoClean methodology introduced) in order to characterize our concepts. We can bind two other types of feature to class concepts and relation concepts. Essential features are those which are usually inherited via generic relation (although we have to provide exception handling capabilities as well, because inheritance is not a necessary requirement), while particular features are not inheritable. Figure 4 shows the most important connections between concepts and features.

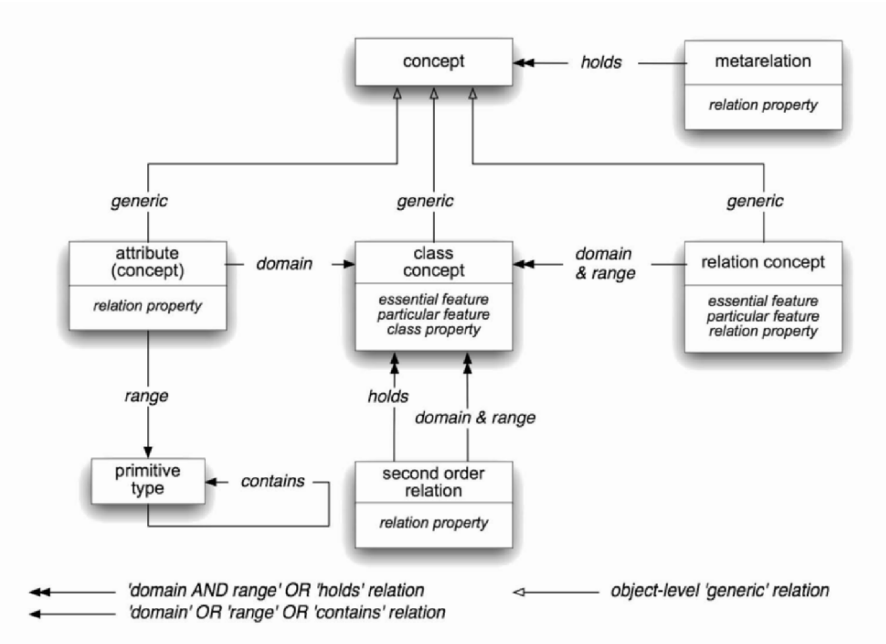


Fig. 4. Connections between concepts and features

5 MEO and OWL

Comparing the MEO concepts to the OWL concepts we can map the most important categories to each other. For example the MEO primitive type has the same meaning and role as the OWL Datatype, and we can MEO attribute to OWL DatatypeProperty, MEO relation concept to OWL ObjectProperty. MEO contains all DL constructors and relation properties which exist in OWL (except those concepts that have individual connections, for example OWL hasValue, because in the first phase we did not implement instance level system components). In MEO we can differentiate ObjectProperty (relation concept) and DatatypeProperty (attribute). Finally axioms are not allowed in OWL which can handle only comments (both exist in MEO), and the concept of feature is completely missing from OWL while it has a very important role in the MEO model.

6 Conclusions

In the MEO project we are building a top level and a domain ontology, but our main focus is on planning, building, using and distributing a robust, consistent ontology infrastructure. In order to ensure the real language independency of our ontologies, we clearly separated the language and the concept levels of our model, and we designed and implemented an integrated and institutionalized set of language management capabilities with which we can easily bind as many languages to our concepts as we wish. This objective and our implementation of it is, in our opinion, unique, we are not aware of any similar projects. While we have built our inner linguistic ontology, we based our work on construction grammar, which is, again, a less frequently used theoretical approach among the ontology building community. Of course, it is usual (maybe necessary), to have the entity of the concept stand in the centre of the model, but the fact, that we differentiated three subtypes of our main entity (concept) and evaluated these as three subordinated concepts with totally equal status is unusual and can be regarded as a novelty. And finally in contrast with the widely applied approach we attributed an important role to relation concepts, and we tried to provide an institutionalized role for features.

References

[Corcho et al. 2003]
Oscar Corcho, Mariano Fernández-López, Asunción Gómez-Pérez: Methodologies, tools and languages for building ontologies. Where is their meeting point?, in: Data & Knowledge Engineering, Vol. 46, 2003, pp 41-64.

[Green et al. 2001]
Green, Rebecca, Bean, C.A., Myaeng, S. Hyon, The Semantics of Relationships: An Interdisciplinary Perspective, Dordrecht: Kluwer, 2001.

[Guarino 2000]
Guarino, N., Welty, C., A Formal Ontology of Properties. Proceedings of 12th Int. Conf. on Knowledge Engineering and Knowledge Management Lecture Notes of Computer Science, Springer Verlag.

[Guarino 2001]
Guarino, N., Welty, C., Supporting ontological analysis of taxonomic relationships. Data & Knowledge Engineering 39 pp 51-74.

[Staab 2004]
Handbook of Ontologies eds. S. Staab, R. Studer, Springer Verlag.

A Semantic-Based Web Service Composition Framework

H-M. Haav*, T. Tammet**, V. Kadarpik*, K. Kindel*, M. Kääramees**

*Institute of Cybernetics at Tallinn University of Technology, Akadeemia 21, 12618 Tallinn, Estonia
**Institute of Computer Science of Tallinn University of Technology, Raja 15, 12618 Tallinn, Estonia

1 Introduction

There is widely recognized problem concerning WSDL-based descriptions of web-services that services are described syntactically not capturing enough meaning [Patil et al. 2004, Probst and Lutz 2004, Elenius et al. 2005]. In order to automate service discovery and composition several semantic heterogeneity problems should be solved.

Most of existing web services are described in WSDL -Web Service Definition Language [WSDL] that is standard protocol for service interface descriptions and SOAP – Simple Object Access Protocol [SOAP]. There is growing effort from the side of Semantic Web research community to make contributions to web service composition frameworks in the form of Semantic Web Services by using specification approaches like OWL-S [OWL-S] or industrial approach called BPEL4WS [BPEL4WS]. As the field of Semantic Web Services is in its early stage of development, then there is lack of practical tools for supporting developers in semantic web service modeling and composition.

Consequently, there is need for conceptual frameworks and tools enriching existing WSDL-based service descriptions with semantic domain-ontology based annotations and providing semantic-based automatic or semi-automatic web service composition for service developers.

The work presented in this paper is aimed at providing semantic-based web service composition framework and a set of supporting tools. The framework takes into account heterogeneity problems of data (input/output) semantics of web services and provides a solution for the problem by introducing domain ontology based annotations to web service WSDL descriptions.

In this paper, we consider only informational web services that do not change the world. This makes our task easier as we do not need to deal with the whole spectrum of semantics of web services, but concentrate only to data semantics. For example, we do not deal with problems of functional and execution semantics of web services as well as we do not consider preconditions and effects.

The main contribution of this paper is in providing framework for introducing domain ontologies describing the concepts of the web services domain, relating them to the data input and output of the web services in order to semantically enrich WSDL-based service descriptions, and providing automatic composition of web services by program synthesis.

A pilot application of the proposed conceptual framework is developed on the basis of Data Exchange Layer X-Road of Estonian State Information System gathering about 700 web-services [X-Road]. The system is implemented for the Linux platform as open source software and is available at [Rql]. As this paper is concentrated to providing the framework, then most of implementation details are omitted.

The rest of the paper is structured as follows. Section 2 presents overall architecture of proposed framework. Section 3 is devoted to semantic annotation of web service descriptions and section 4 describes a logic-based web service composition component of the framework. Section 5 is devoted to related works and section 6 concludes the paper.

2 Architecture of the framework

In this section, an overall picture of proposed framework for composing web services is given and explained (see Fig. 1). The architecture of the framework consists of 2 interrelated parts: web service annotation and logic-based web service composition components. Goal of the system is to automatically find a plan for service composition as an answer to the user request.

The most important processes of the framework are as follows:

1. Building web services domain ontology by using ontology editor Protégé 3.1.1 [Protégé 3.1.1] and creating corresponding OWL description of the ontology.
2. Annotating existing web services described in WSDL on the basis of created ontology by mapping concepts from ontology and labels of WSDL descriptions.
3. Taking as input the user requests for composition of new services. The user request is expressed on ontology level i.e. described in ontology terms.
4. Translating the user requests (queries), domain ontology descriptions in OWL as well as annotated WSDL descriptions of web services to the logic language ELM [Tammet 2004a] in order to make them understandable for the logic-based component of the system.

5. Performing reasoning tasks and composition of web services by program synthesis using the theorem prover RqlGandalf [Rql] that is extended version of Gandalf [Tammet 1997].
6. Extracting the result of program synthesis as a Python program corresponding to the required composite service.
7. Developing the composite service by refining and further implementing the synthesized Python program.

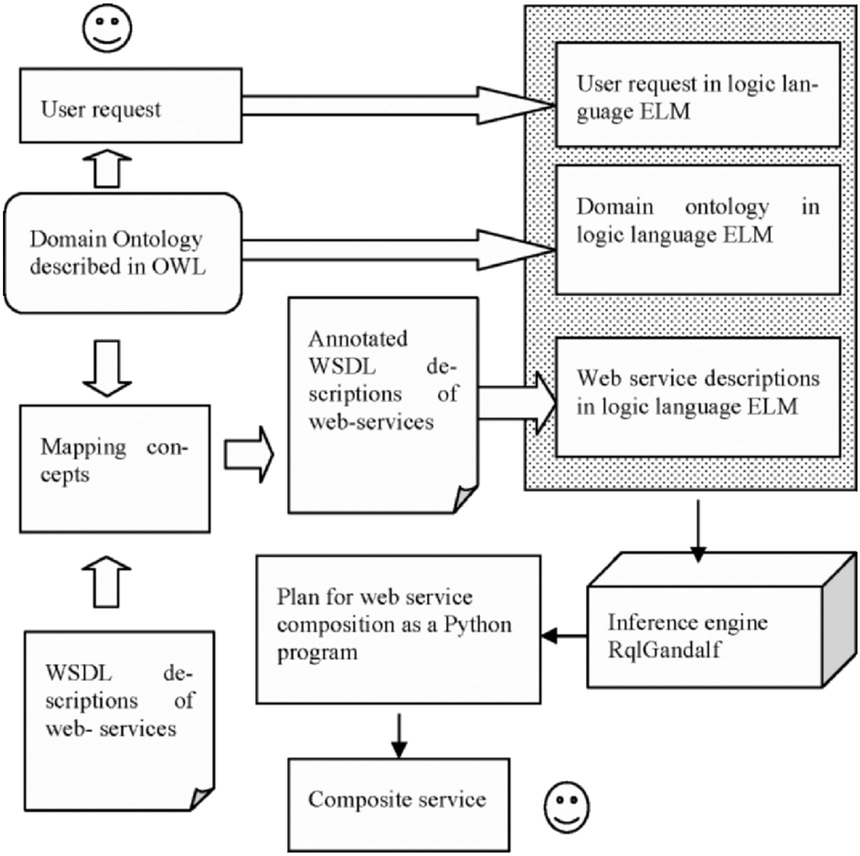


Fig. 1. Web-service annotation and composition framework

We now describe both components of the framework: web service annotation and logic-based web service composition components in more detail as follows.

3. Semantic annotation of web service descriptions

In this paper, we deal with data semantics of web services and do not consider other categories of semantics of web services like functional and execution semantics as identified in [Patil et al. 2004].

3.1 Data semantics of web services

According to our analysis done for Data Exchange Layer X-Road [X-Road], we identified semantic problems of data input and output descriptions of services captured in WSDL-based service descriptions. Similar observations can be found in [Probst and Lutz 2004] for GIS.

In this paper, we consider two types of semantic problems of WSDL-based service descriptions as follows:

1. Labels of input and output data types of services are different, but the descriptions denote the same thing or have the same meaning. It is clear that this type of semantic problem makes it hard to meaningfully compose new web services and makes it impossible to automatically compose web services.
2. Labels of input and output data types are the same, but the descriptions do not denote the same thing or do not have the same meaning. This heterogeneity can lead to incorrect composite web service, as the user or a program does not know that the identical labels in different services refer to different domain concept.

Annotating WSDL descriptions of service input and output with concepts from domain ontology can solve the semantic problems discussed above. This prevents that the services having conceptually not matching input and output are combined in a composite service. There is the third type of semantic problem related to data semantics also identified in [Probst and Lutz 2004] for GIS: type mismatch problem. In this paper, we do not deal with this problem; even we have identified this problem in X-Road web-service descriptions. In our approach, the user should resolve this for each case of type mismatch.

3.2 Semantic annotation

Our goal in developing service composition framework for X-road was to annotate existing web services already described in WSDL. We rely on manual annotation approach as most of developers, because research on automatic or semi-automatic annotation is in its early stage. Nevertheless, there are good examples like the Meteor-S framework for semi-automatic annotation of web services with ontologies [Patil et al. 2004], but these are not yet practical implementations.

According to our approach, a simple web based editing tool could be used for mapping input/output of web services WSDL descriptions to concepts from a domain ontology described in OWL. We used Protégé 3.1.1 ontology editing tool [Protégé 3.1.1] for creating domain specific ontologies and converting them to OWL. In the following we demonstrate our semantic annotation approach on the basis of an example from our pilot application.

3.3 Sample domain ontology

Let us consider small sample ontology from the state information system capturing concepts of citizen and enterprise. Most of web-services of the state information system are related to the first concept. Citizen concept has the following properties: ID-code, First_name, Last_name, Citizenship, and Place_of_birth. Citizen concept has 2 subconcepts as Estonian_citizens and Foreigners. Enterprise concept has the following properties: Reg_no, Name, Field_of_business, and Owner. Enterprise concept has also 2 subclasses: State_owned and Private.

3.4 Sample web services

Let us suppose that we have the following sample set of individual web services available as follows:

1. Web service GetPersonalData finds all personal data of a citizen by citizen's identification code. Its input data type has label *idcode* and its output is complex type having the following sequence of elements in the corresponding WSDL file: *idcode, fname, lname, citizenship, place_birth*.
2. Web service GetCompany finds among other output fields a registration number of a company by business field of this company. Its input parameter has label *businessfield* and one of its output fields is denoted by label *reg-code* in the corresponding WSDL file.
3. Web service GetOwner finds identification code of a person, who is owner a company by given registration code of a company. Its input parameter has label *comp_reg_no* and its output is denoted by label *owner_id_code* in the corresponding WSDL file.

3.5 Annotation of service descriptions

Web services are described using WSDL and annotated using the ontology as represented above. As input for the annotation process are ontology description in OWL (produced by Protégé) and web services input and output parameters' descriptions from WSDL file. We need to annotate only input/output of web services, as our service composition component needs this conceptual information for composition process.

For example, *idcode* as the input parameter name of the service GetPersonalData should be mapped to the attribute ID-code of the concept Citizen in the ontology above. The output of the same service should be mapped as follows: label *citizenship* to the attribute (slot) Citizenship of the concept Citizen in the ontology, label *fname* to the concept *First_name*, etc. Corresponding similar mappings should be done for input/output of other web services as well. The described above process of semantic ontology based annotation of web services described by WSDL files is semi-automatic. Using annotation editor, the user can load WSDL file to be mapped to suitable domain ontology.

Another approach might be to build the editor into Protégé as in [Elenius et al. 2005], where in contrast to our approach; OWL description of domain ontology is enhanced by instances that refer to corresponding elements (syntactic names of input/output) from WSDL descriptions.

4 A logic-based web-service composition component

4.1 Composition of web services by program synthesis

Our logic-based web-service composition approach is based on general scheme of automated deduction and program synthesis [Green 1969, Manna and Waldinger 1980]. According to that, we formulate an axiomatic theory of the application domain including background knowledge. In this theory, domain ontology concepts and their properties are expressed by axioms. Each web service is expressed by a set of axioms that describes the service. The query (request) is expressed as a theorem in the language of application domain theory. Usually, the theorem expresses the existence of the entity (e.g. a citizenship, a name, etc) that the query is asking to find by given entity or entities. Using an automated theorem prover, it is proved that the theorem follows from the axioms of the application domain theory. The proof should be sufficiently constructive, so that from proving the existence of a desired entity we can construct a program of finding it.

As the automated theorem prover we use FOL theorem prover Gandalf [Tammet 1997] extended for synthesis of programs in order to be suitable for web service composition tasks [Rql]. The core method employed by the system is the classical combination of the resolution method with the ans-predicate mechanism; see [Green 1969, Tammet 1995]. The system is looking for suitable instantiations of the variable(s) in the query. These variables will be the arguments of the special ans-predicate. A clause containing only the ans-predicates is considered to be an empty clause, with the arguments of the ans predicate representing the solution to the query.

The simplistic usage of the ans-predicate will not be sufficient for program synthesis. We will need additional rules and mechanisms. The four main extensions built into the core prover are as follows:

1. Built-in strings, dates and arithmetic.
2. A PROLOG-like mechanism for deriving different alternative answers.
3. A derivation rule for eagerly disambiguating clauses containing several answerliterals.

4. A derivation rule for converting logical conditions into term-level conditions. For example, if we have a clause —A|ans(s), then we can derive ans(if (A, s)) as a new clause, provided that A is computable. Resolution proof search extended by such rules may give us a result in the form of a term. Next, this term should be compiled to an executable program.

Our compilation stage consists of two phases as follows. First step is the loop-lifting conversion modifying the term. Second step constitutes compilation of the modified term to an executable Python function. We have decided to use Python for practical usability, and we are also avoiding comprehension and lambda-terms in the Python program.

An important methodological decision we have taken is to avoid any forms of explicit induction. We describe lists using special function symbols, which will eventually be translated to loop constructions. Similarly, we avoid error handling in logic. The synthesized programs will handle possible error situations instead.

4.2 Methodology

As any other logic-based system, our system also can accept different input specification languages and convert them to logic for reasoning purposes. We have developed a user-friendly rule language ELM [Tammet 2004a] to be used and understood by developers (users). This language is built on the top of core language ECL [Tammet 2004b], which is basically first order logic extended with several data types like numbers, string, dates, etc. and operations on them. ECL is an input language for the theorem prover Gandalf.

According to our framework, user requests (queries), domain ontology descriptions in OWL as well as a part of annotated WSDL descriptions of web services are converted to ELM language (see Fig.1) in order to make them understandable by logic-based component of our system.

For capturing ontological information we need OWL to ELM converter and for getting descriptions of input and output data types of web services we need extractor of this information from annotated WSDL descriptions and converting the latter to ELM language.

Our intention is that the end user operates on the level of domain ontology concepts and places his/her requests on this level.

In the next section, we give main principles of conversion of domain knowledge represented in OWL and annotated WSDL to logic language in order to form domain theory for program synthesis. Due to limited scope of the paper we need to omit exact definitions of transformations from OWL to ELM.

4.3 Transformations to the logic language ELM

The ELM language [Tammet 2004a] is a sugared version of extended FOL. Extensions to FOL allow writing predefined computable functions, sentences in SQL and several nonmonotonic and (weak) epistemic operators, as well as select data sources to be used in queries.

In this chapter, not full syntax of ELM is presented, but it is described without any formal definitions of syntax or semantics in the scope needed for explaining examples. The formal definitions for the language can be found in [Tammet 2004a].

Taking into account features of ELM we now demonstrate translations from OWL to ELM and from annotated WSDL to ELM. In the latter case, only part of WSDL description is needed for logical reasoning.

Translating OWL ontologies to ELM

Translation of OWL ontologies to FOL axioms is not new as correspondence between OWL and Description Logic (DL) is well known [Baader et al. 2003] as well as translation from DL to FOL [Borgida 1996]. As ELM language is extended FOL, then the translation of OWL ontologies to ELM can be done in standard way.

In the following examples, OWL ontology from section 3 is translated to ELM in order to demonstrate our transformation principles.

- ***Representation of classes.*** We use unary predicates for representing classes. In the ELM language we may write, "X is something" to say that X is something (in standard logical notation, something(X)), where X is either OWL individual name or ELM variable. Variables are denoted by symbol "?" in ELM language. OWL classes *Citizen* and *Enterprise* are translated to the following unary predicates*:?P is citizen,?E is enterprise*
  Each OWL instance is then translated to literal fact like
  John is foreigner or *Jaan is citizen* or *IBM is enterprise*
- ***Representation of properties.*** Each OWL property R is translated to binary predicate R(X,Y). For example, if OWL class *Enterprise* has a property businessfield, then this is translated to the following ELM sentence*: ?F is businessfield of enterprise*
- ***Representation of subsumption relationships.*** Each subclass/superclass (subsumption) relationship is translated to the following rule in the form: *Each predicate1 is predicate2*
  For example, *Each estonian_citizen is citizen*
- ***Representation of queries.*** User requests (queries) are given on ontology level and translated to ELM as following sentence: *Find var_1,...,var_N where sentence*?, where *var_1,...,var_N* are variables and *sentence* is an ELM sentence.
  For example, our sample request is translated to the following ELM sentence:
  find ?S where banking is businessfield of ?C
  and ?O is owner of ?C and ?S is citizenship of ?O

Web service description in logic language

In order to translate web service descriptions to ELM language annotated WSDL to ELM translations are needed. Not entire WSDL description is converted to ELM, but necessary information about data input and output of the service and corresponding service name are extracted from the WSDL file and converted to a sentence in logic based language ELM.

For example, previously given web services are expressed in ELM language as follows.

If a web service returns as output a structure, then for each ouput field of the web service the following type of sentence is created as for example for GetPersonalData returning a structure containing different data fields of personal data according to the properties of the concept of citizen.

if ?I is ID_code of ?P then
getfield(citizenship,call(GetPersonalData,?I)) is citizenship of ?P.

If a web service has only one output field as GetOwner, then the following typical sentence is generated.

if ?R is Reg_no of ?C then
call(GetOwner,?R) is ID_code of ?P.

The web service GetCompany returns a list of companies operating in a given business field. As Reg_no is one output field of GetCompany, then the following sentence is created.

if ?F is businessfield of ?C then
foreach(L,getfield(Reg_no,L),call(GetCompany,?F)) is Reg_no of ?C.

In general, this type of sentence is generated for each output field.

4.4 Composition of web services

Given web services domain ontology and web services descriptions in ELM language, the logic-based component of our framework translates these to internal logic language of FOL theorem prover Gandalf (i.e. to extended FOL). The resulting theory is passed to Gandalf, where it can be used for reasoning tasks and for composition of web services by program synthesis as discussed in the beginning of this section. We now show the web service composition on the basis of our running example.

Let us suppose that the user (service developer) requests to compose a new service that finds a citizenship of owner(s) of an enterprise by given business field of an enterprise. This request is simplified but rather similar to what is used in real application. In order to satisfy the request the composite web service should be created as follows.

If the user is submitting the query to find citizenship of owner(s) of an enterprise, which business field is banking, then this query is translated to the following sentence in the logic language ELM:

find ?S where banking is businessfield of ?C
and ?O is owner of ?C and ?S is citizenship of ?O

The system tries to find an answer to this query and synthesizes a program that implements corresponding web service. The resulting program is generated in Pyhton programming language as follows:

```
def query( ):
    result=[ ]
    data=call("GetCompany",”banking”)
    for i in data:
        result=result+\
        [getfield(“citizenship”,call(“GetPersonalData”,call(“GetOwner”, i)))]
    return result
```

The result is a list of citizenships of owners of enterprises, which operate in banking field. This automatically obtained program code is presented to the developer of web services in order to aid him/her in the web service composition process. This code needs further refinements, but in principle, having individual web services available, it can invoke them in the sequence given by the synthesized program.

4.5 Implementation

A proof of concept implementation of the web service composition framework is done as a part of the larger system of development of web-based applications called Rql technology [Rql]. The technology consists of 2 main parts: the application server Xstone for creating 3-layered systems and the RqlGandalf solver. The middleware server Xstone connects to Oracle, PostgreSQL databases and the RqlGandalf solver. In addition, the Rql technology includes browser-based interface creating toolkit for Xstone. The whole system can be downloaded from [Rql]. It is written for Linux in C and licensed under GPL.

5 Related Works

Semantic markup of web services has been proposed by many projects and initiatives like WSDL-S [WSDL-S], WSMO [WSMO], OWL-S [OWL-S], METEOR-S [Patil et al. 2004]. Semantically most expressive is OWL-S, but OWL-S profile model duplicates input and output descriptions embodied in the WSDL used for grounding of OWL-S descriptions. It is also bound to only one ontology language OWL. OWL-S being web service ontology does not really solve the data semantics problems discussed in the beginning of this paper as it is intended to semantically describe all aspects of web services including composition. Data semantics problems in OWL-S should be solved on the WSDL level that leads us in any case to WSDL annotation. These are reasons why our approach is rather close to WSDL-S. Comparing to WSDL-S, our semantic annotation approach is not as general as WSDL-S proposal. We have used very pragmatic solution of annotating only input/output of web services in order to provide (semi)-automatic web service composition framework for service developers. Our approach is independent on the service description language in its logic based service composition part.

Regarding to automation of web service composition, the approaches proposed fall into two main categories based on workflow models or on AI planning. Our work is tightly related to AI planning approaches based on automated program synthesis that relies on strong theorem proving technology. In [Waldinger 2001, Waldinger and Shrager 2006] available services and user requirements are described in FOL, and then constructive proofs are generated with SNARK theorem prover. Nevertheless, our approach differs from their approach in that they do not consider web services and do not relate their approach to web services description standards. We also enhanced FOL theorem prover in order to meet requirements of synthesis of web services.

Other logical approaches are used for web service composition in [Rao et al. 2004, Ponnekanti 2002, Sirin et al. 2004]. In [Rao et al. 2004] linear logic (LL) is used. For external presentation of web services they use semantic web service language DAML-S and for composition process they translate web services into extralogical axioms and proofs in LL. Service composition tool SWORD [Ponnekanti 2002] generates composite service plans by using rule-based plan generation implemented in Java. [Sirin et al. 2004] have developed semi-automatic service composition prototype, which consists of two basic components: a composer and an inference engine. First user selects the service he/she is interested in, and then inference engine finds all the other services that can supply appropriate data for selected service input.

6 Conclusions

The paper presented a conceptual framework for semantic-based web service composition that consists of 2 interrelated components: semantic-based web services annotation and logic-based web services composition components. We have shown how WSDL descriptions of existing web services have been annotated on the basis of introduced domain ontologies and converted to logic based language for logic-based service composition. We have demonstrated how FOL theorem prover RqlGandalf has been used for automatic composition (synthesis) of annotated web services.

Many of theoretical and practical aspects of the program synthesis are still open. Our system is not always capable of synthesizing a program, nor is it capable of selecting a best variant from several different synthesized programs. This constitutes our future work.

7 Acknowledgements

This work was partially funded by Enterprise Estonia funding within R&D project RQL "Rule-based databases for creation of web services" and ESF grant 5766. We thank anonymous referees for their comments.

References

Baader F, Calvanese D, McGuinness D, Nardi D, Patel-Schneider P (Eds) (2003) The Description Logic Handbook, Cambridge University Press

Borgida A (1996) On the relative expressiveness of description logics and predicate logics, Artificial Intelligence, 82(1-2): 353-367.

BPEL4WS, Available from: http://www-128.ibm.com/developerworks/library/ws-bpel

Elenius D, Denker G, Martin D, et al (2005) The OWL-S Editor- A Development Tool for Semantic Web Services, Gomez-Perez, A., and Euzenat, J. (Eds) Proceedings of ESWC 2005, LNCS 3532, Springer-Verlag, pp 78-92.

Green C (1969) Application of theorem-proving for problem solving. In Proceedings of 1st International Joint Conference on Artificial Intelligence, pp 219-239.

Manna Z, Waldinger R (1980) A Deductive Approach to Program Synthesis, ACM Transactions on Programming, Languages, and Systems.2: 90-121.

OWL, Web Ontology Language. Semantics and Abstract Syntax. W3W Working Draft 31 March 2003. Available from: http://www.w3.org/TR/owl-semantics/

OWL-S, Web Ontology Language for Web Services. Available from: http://www.daml.org/services/

Patil A, Oundhakar S, Sheth A, Verma K (2004) METEOR-S Web service Annotation Framework, Proceedings of WWW Conference 2004, ACM, New York, pp 553-562.

Ponnekanti SR, Fox A (2002) SWORD: A Developer Toolkit for Web Service Composition, Proceedings of Eleventh World Wide Web Conference (WWW2002, Web Engineering Track), Honolulu, Hawaii

Probst F, Lutz M (2004) Giving Meaning to GI Web Services, J. Hu (Ed), Proceedings of Second International Workshop on Web Services: Modeling, Architecture and Infrastructure, Porto, INSTICC Press, pp 23-35.

Protégé 3.1.1, Protégé OWL Editor. Available from: http://protege.stanford.edu

Rao J, Küngas P, Matskin M (2004) Logic-based Web Services Composition: from Service Description to Process Model, Proceedings of the 2004 IEEE International Conference on Web Services, ICWS'2004, San Diego, California, USA, July 6-9, 2004, IEEE Computer Society Press, pp 446-453.

Rql, Rule-based systems for creation of web services. Available from: http://deepthought.ttu.ee/it/xstone/

Sirin E, Parsia B, Wu D, Hendler J, Nau D (2004) HTN planning for web service composition using SHOP2. Journal of Web Semantics, 1(4): 377-396.

SOAP, Available from: http://www.w3.org/TR/soap12-part0
Tammet T (1995) Completeness of Resolution for Definite Answers. Journal of Logic and Computation 5(4): 449–471.
Tammet T (1997) Gandalf. Journal of Automated Reasoning 18(2): 199-204.
Tammet T (2004a) Extended Logical Markup language (ELM) Model and Syntax Specification. Available from: http://deepthought.ttu.ee/it/elm/elm.html
Tammet T (2004b) Extended Common Logic (ECL) Model and Syntax Specification. Available from: http://deepthought.ttu.ee/it/elm/ecl.html
Waldinger R (2001) Web Agents Cooperating Deductively, Rash, J.L. et al (Eds): Proceedings of FAABS 2000, LNAI 1871, Spriger-Verlag, pp 250-262.
Waldinger R, Shrager J (2006) Deductive Discovery and Composition of Resources, Proceedings of RoW2006: Reasoning on the Web, Edinburgh, Scotland
WSDL, Available from: http://www.w3.org/TR/wsdl
WSDLS, WSDL-S: Adding semantics to WSDL - White paper. Available from: http://lsdis.cs.uga.edu/library/download/wsdl-s.pdf
WSMO, Web Service Modeling Ontology. Available from: http://www.wsmo.org/
X-Road, Data Exchange Layer X-Road. Available from: http://www.ria.ee/27309

Community-Based Partnerships in the Design of Information Systems: The Case of the Knowledge Commons

Natalie Pang, Henry Linger, Don Schauder

{natalie.pang, henry.linger, don.schauder}@infotech.monash.edu.au

Abstract: Using the case of a cultural institution in Australia, the paper introduces the context and application of the knowledge commons, a vision in cultural institutions and in doing so, argue for the case of community centred approach to design based on structurational and action research principles for cultural communities. The paper argues for the application of design principles and in studying how these principles may be enacted in cultural institutions; it argues for potential of empowering and sustaining communities.

1 Introduction

In cultural institutions, such as museums, galleries and libraries amongst others, a knowledge commons represents the process of creating and sharing public knowledge. In this public domain the design of information systems is characterised by changing structures of ownerships that raises issues of power, the intentionality of tasks (between individuals, groups, organisation, and the context of the societal system) and the control and flexibility expressed in the design. This allows a focus on the users changing perceptions of the information systems and the relationship between the variety of actors within the design process and their relationship to the information system design.

The importance of understanding users was first articulated by Norman and Draper (1986) who coined the term user centred design (UCD) that 'emphasizes that the purpose of the system is to serve the user, not to use a specific technology, not to be an elegant piece of programming. The needs of the users should dominate the design of the interface, and the needs of the interface should dominate the design of the rest of the system' (Norman, 1986). In this sense, the interface functions more than just a medium between the computer and the user; it also acts as a type of intermediary communicator, negotiating understandings and meanings between the computer and user. For cultural institutions, an information system is an emergent property of a dynamic ecosystem and the design process must incorporate a cultural vision and accommodate community engagement based on structurational and action research principles.

This paper explores how UCD principles are enacted in the process of creating and sharing public knowledge in cultural institutions. This involves the investigation of UCD concepts and their application to the ongoing development of cultural institutions as multi-stakeholder partnerships. We examine the discourses around user-centred design as an approach for the design of the information system and extend these to accommodate the specific requirements of the knowledge commons. The link between the design of information systems and its place in the knowledge commons is discussed from insights gained from the case of a unique relationship between a women on farms' community and a museum. From these insights we argue to extend UCD as a community-based participatory design approach for the purpose of contributing to the construction and sustainability of the knowledge commons in cultural institutions. The paper demonstrates how this approach to the design of technological applications and systems serves as a dynamic influence in facilitating the knowledge commons in cultural institutions and their communities.

For this study we seek to define 'museum' but this is not intended to be an exhaustive definition. According to the International Council of Museums (ICOM), a museum is 'a non-profit making, permanent institution in the service of society and of its development, and open to the public, which acquires, conserves, researches, communicates and exhibits, for purposes of study, education and enjoyment, material evidence of people and their environment.'

2 User-Centred Design: A Complex of Disciplines

A premise underlying UCD is that the dynamic interplay of user needs and feedback and translating these needs into some practical specification of requirements for an information system cannot be satisfied in a linear, non-recursive methodology of design. The difficulty users have in communicating ideas or concepts (Bonner, 2002) ensures that capturing user needs is time-consuming and costly. The iterative view of UCD implies that there can never be an "ideal" information system for the community to use (Fraser, 2002; Head, 1999; Cockton, 2004). But perhaps this is not a problem to be solved but an opportunity for information systems requirements to reflect the continuous negotiations between cultural institutions and communities.

Perhaps one of the most famous representations in thinking about designing around users came from Norman's (1988) groundbreaking ideas in 'The Psychology of Everyday Things' reflecting a paradigm shift in information systems design. It is not our intention to critique information systems design methods but to focus on how the user is incorporated into the design process.

The main contributions to UCD advocated by well known researchers in this area are summarised in Table 1. This table is not intended to be an exhaustive, but highlights some of the most significant contributions towards the study of UCD since 1983. Many of the works in Table 1 result in methodological instruments that are applied to users who are assessed individually, seated in front of their computers.

Our focus on designing with the user, instead of designing for the user, allows participatory design to be interpreted as UCD. This interpretation involves the collective participation of all stakeholders including communities. Moreover, participation in the design process is not seen as a once-off assessment of stakeholder and community needs, but as ongoing dialogues that are both explicit and implicit.

Author(s)	Main area(s)	Contribution
Carroll, J (2000)	Scenario based design and claims analysis	Introduced the idea of iterative design through the task-artifact cycle, and user-centred design strategies in scenario based design and claims analysis. Carroll (2000) stresses the importance of maintaining a continuous focus on situations of and consequences for human work and activity to promote learning and the structure and dynamics of problem domains, thus seeing usage situations from different perspectives. His work is significant in the field of process-oriented design modelling.
Nielsen, J (1993)	Ten usability heuristics	Main contribution in using heuristic principles as a method for usability evaluation, to be used in any stage of a development process.
Schneiderman, B (1992)	Nassi-Shneiderman diagrams, and the 'Eight golden rules of interface design'	The nassi-shneiderman diagram was co-developed by Shneiderman and Nassi, a graphical design representation for structured programming: breaking large tasks into smaller subtasks, until only simple statements remain. The main contribution of this work is a top-down approach of design. This work is rarely used nowadays, as modifications usually require the whole diagram to be drawn. His other main contribution in the field lies in propagating rules of interface design, similar in nature to Nielsen's usability heuristics.
Norman, D (1988)	The Psychology of Everyday Things	Norman was perhaps one of the earliest to apply insights from the field of psychology and industrial product design and apply them to the design of user interfaces. He also introduced the model of interaction, a framework to explain human interactions with systems.
Card, Moran and Newell (1983)	GOMS	The GOMS (goals, operators, methods, selection rules) method is a model used for task analysis to analyse and predict total time for user task performance. Alongside with this purpose, GOMS also makes clear a hierarchy of goals and sub-goals of users.

Table 1. Key contributions in UCD

2.1 The User in Design

A historical analysis of the role of the 'user' highlights the shifts in the perceptions and roles of the user in design. The term UCD is widely used, but there appears to be only a basic consensus as to its meanings and implications: user needs should inform the information systems design process. Although organisations express their own UCD philosophies in different ways, all would claim a focus on the functional needs of users. Karat and Karat (2003) acknowledge the diversity of interpretations of UCD, but note that they are all agreed in distancing themselves from Taylorist principles of techno-centric, mass production in information systems design methodologies. Beyond these clear points of agreement there is a broad consensus in the UCD world that reality is 'mutable', there are 'no certain truths', and 'knowledge is constructed through communally created knowledge and action'. Gulliksen et al. (2003) further reinforces this alarming observation in that 'the concept of user-centred…design has no agreed upon definition'. Much of the research in UCD is driven by a concern that the lack of consensus is obscuring the concept of UCD, turning it into a concept with no pragmatic meaning. This gives rise to misunderstandings about the effectiveness of UCD.

A significant contributed to the conceptual development of UCD has been the establishment of ISO 13407, an international standard established in 1999. The standard aims to provide 'guidance on human-centred design activities throughout the life cycle of computer-based interactive systems' (Jokela et al., 2003). Almost as if agreeing to the observation that there is simply too much variety in UCD methods and techniques, ISO 13407 is set up for 'those managing design processes' (Jokela et al., 2003) and does cover methods and techniques.

The user as 'a user'

The term UCD, as introduced by Norman and Draper (1986), account for many of the contributions outlined in Table 1, where key research areas have included cognitive engineering and psychology, interaction models between the individual user and computer, the development of heuristics, and so on. These activities typically involve testing by seating users in front of individual computers with testing being conducted on individuals or groups. Another significant activity lies in gathering of user requirements. Requirements are usually collated from the assessments of needs collected through meeting with representatives from the recognized user communities.

While such efforts are worthwhile, it challenges the UCD concept by characterising the user as a 'victim' to be 'rescued' (Spinuzzi, 2003). Such an interpretation can be drawn from UCD literature that concentrates on the adeptness of designers, developers, usability specialists and managers to capture the needs of users effectively. But these approaches go no further than providing effective interfaces and systems to 'rescue' end users. Such passivity results in a lack of understanding about the decisions facing community groups, and the barriers and challenges involved (Merkel et al., 2004).

From user to participant

A principle advocated in participatory design (PD) is that UCD involves the collective participation of all stakeholders (Schuler and Namioka, 1993) and is distinguished from other UCD approaches in its philosophy of inclusion of the user as early as the conceptual stage of design. PD approach is beyond designing around the interface as it views users as participants in the design process and shifting responsibilities, such as prototyping, to the users (Gulliksen et al., 2003). The developments in PD have much to thank the Scandinavian efforts in their advocacy for extensive user involvement in design (Gulliksen et al., 2003; Taxen, 2004). Taxen (2004) noted that the motivation in the majority of the early PD projects in Scandinavia was an agenda of empowering workers when technologies were introduced in the workplace. This empowerment and transposition of power and responsibilities is a significant aspect of the design process that is central to community based design as argued below.

Community-centred design

According to Preece (2000), the community-centred design is both participatory and evolutionary. The design of socio-technical environments is the main focus in community-centred design, with the emphasis on social interactions between humans mediated by technology instead of human-computer interactions. Here the social elements that influence the use of technology and interactions amongst users are addressed (DePaula, 2003).

Arguably, community-centred design is included in the domain of socio-computing, a recent research area concerned with the influence of technology on socio interactions and vice versa (DePaula, 2003). Socio-computing has been viewed through the lens of structuration theory (Giddens, 1986), adapted in the context of design (Pang and Schauder, in press; Pang et al., 2006). Structuration theory is used as a framework for design in this discussion because of its ability to recognise both the duality of structure and emergence of use that is a typical outcome of this design process as has been appropriately highlighted in the literature (Pang and Schauder, in press; Pang et al., 2006; Orlikowski and Robey, 1991; DePaula, 2003). In addition to structuration theory, the community-centred design process is also an interventionist approach that is informed action research principles. The combination of structuration theory and action research principles ensure that community centred design (CCD) empowers communities and leads to the co-ownership, co-construction, and co-production of dynamic design, systems, and information resources. These are the outcomes that are the focus of cultural production in the knowledge commons that cultural institutions seek.

The knowledge commons

The idea of a commons is not new – in fact it has been around as long as the first human cooperation. Men hunting together for food, and sharing their skills and eventually, their produce, the commons, is rooted in communities of social trust and cooperation (Bollier, 2004). Derived from the historical commons, the commons defined by Benkler (2003) are 'institutional spaces, in which we can practice a particular type of freedom; freedom from the constraints we normally accept as necessary preconditions of functional markets'. The term has also been used to refer to the infusion of digital technologies and resources to be used freely (MacWhinnie, 2003; Hales et al., 2000; Beagle, 1999; Bailey and Tierney, 2002; Bollier, 2004; Cowgill et al., 2001; Lukasik, 2000). Moritz (2004) defines the knowledge commons as 'zones of free and equitable use for data, information and knowledge' consisting of physical, logical and content layers of resources (Bollier, 2004).

It should be highlighted that several understandings of the 'commons' have emerged with the popular ones being the information commons, the learning commons, and the knowledge commons. The information commons emphasises the free and equitable use of information resources, while the learning commons (an increasingly popular concept used by libraries) infuses services, resources and technologies in spaces to promote learning. The focus of the learning commons is on the open nature of processes to capture revelations from information and knowledge; to learn, in other words.

In this paper we focus on the knowledge commons that is envisioned to include not only information resources, but also indigenous knowledge and processes and communities that concern themselves with the production of knowledge. Such a purpose implies that these communities come together not only for the creation and sharing of information resources, but also involves processes, explicit or implicit, that facilitate the communities' production of knowledge. This conceptualisation of the knowledge commons is critical as information, as a scarcity, could be deplete while knowledge, as a resource, 'has the characteristic of not being degraded when used, but rather to increase in value' (Drucker, c.f. Hellstrom, 2003). This also recognises that communities in the knowledge commons are dynamic and their creation and sharing of knowledge can be influenced by their very dynamism.

A key principle of the knowledge commons lies in the perception of people; they are seen not just as users, or even participants, but also as creators and co-producers. The relationship between the UCD approach and the knowledge commons lies in the commonality they share in the inclusion of dialogue from the community: individual voices, group reflections and influences from institutions and the larger society. The implication for UCD is that design consists less in products than in processes that engage all participants in reflective practice and continuous learning expressed in negotiated action outcomes.

In this paper we looks at the knowledge commons in the context of cultural institutions. These institutions seek to protect, preserve and construct collective resources, and in doing so, make resources freely available to their communities. Today, these institutions are constantly challenged with privatisation and market forces seeking to dominate even the information and resources they are protecting and creating. One example is the licenses-limited access to electronic periodicals and journals. The core role of cultural institutions, their raison d'etre, has not changed but the means by which this role is discharged has changed radically.

Charles Leadbeater recently said that if we neglect the public platforms of our society then we are neglecting a vital part of what makes us a society (Leadbeater, 2006). Cultural institutions form a large part of this public platform. In addition, on this platform they ensure the creation, sustainability, preservation, and convergence of communities and their cultures.

Through a discussion of a museum case study and its relationship with a community, we demonstrate how cultural institutions create and sustain a knowledge commons in the community, in zones of free and equitable access and use, amidst technological and environmental changes, using as a starting point and a continuing motif, a collaborative endeavour in design.

Addressing contexts through tasks

In considering community centred design, there needs to be a framework for considering various interactions of communities. Thus tasks need to be considered beyond the traditional methods of HCI (human-computer interaction) that focus on the work level task. In our framework we are particularly interested in addressing the contexts of tasks to reveal the real problems and needs. At the same time, our the focus on communities and the commons necessitates that UCD issues are elucidated at both the individual and group levels. The concept of community is bound up with various kinds of group affiliations or belonging by individuals. Consequently, CCD needs to reflects an understanding that individuals belong to multiple communities.

The focus for a renewed vision of UCD, CCD, is on the interaction between the individual and the group, and the community culture generated by this interplay against the backdrop of wider social structuring processes within the knowledge commons. Linger (2002) arrived at a model, Figure 1 below, that extends the task-based approach to Knowledge Management (Burstein and Linger, 2003; 2005) across individual to societal levels of analysis and engagement. The formulation of this model is influenced by the Information Continuum Model (ICM) (Schauder et al., 2005) that is constructed on key structurational concepts from Giddens as applied to the creation, capture, organisation and pluralisation of information at the levels of individuals, groups, organisations, and societies (Schauder et al., 2005).

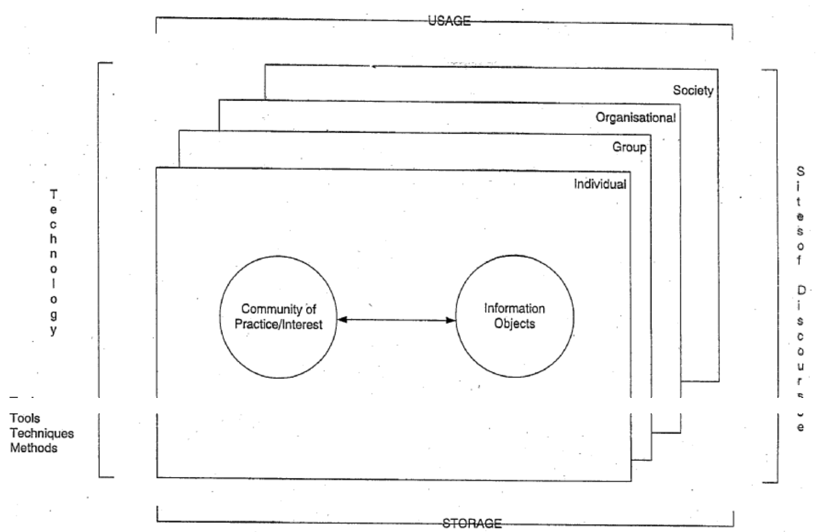


Fig. 1. The Linger model (Linger, 2002; Pang and Schauder, in press)

Conceptualising this model in the context of CCD highlights the following:

Communities and information objects

Referring to the 'common resources' within the knowledge commons, information objects are defined as the public resources cultural institutions seek to provide for communities. Depending on the cultural institution, this may include literary works, digital resources, spaces, moving pictures and other multimedia, community information, and so on. These information objects come about through the construction and reconstruction of cultures by communities. They have both structural and agency relationships with communities, and vice versa.

Layers/levels of interactions

The interactions between communities and information objects are contextualised using 4 layers of interpretations. This is an extremely important as information objects that have not been contextualised have no agency attributes. Within the *individual* layer, the individual makes sense of her own self-knowledge and engages in private projects. This is an important construction of knowledge that is later shape, and in turn be shaped by, other layers of interpretations.

With this construction of knowledge, the individual contributes to the production of knowledge within the community, the *group* layer in Linger's model. This is in turn shaped by organisational influences that impact on the communities and their interactions with information objects. However, the organisation are also significantly influenced by the very dynamics of the interactions at the community level. The fourth and overarching layer lies in societal interpretations that shape the organisation while at the same time are collectively shaped by the organisation and its partners.

These layers of interpretations relates also to two ends of a spectrum of information systems. On one end is the storage of information objects and the other relates to the usage of information objects in the context of the knowledge commons. For the purpose of this study, the usage-end of the spectrum is the focus of our study.

Concluding remarks

Tasks in communities are considered against these layers of interactions, and are considered through the lens of cultural institutions. Having this in mind, it must be noted that the four levels of interactions (and tasks): individual; group; organisational; and societal; are not dichotomies but dynamically related within their contexts. The mapping of tasks in design must be cognisant of these levels and take into account the interplay between communities and information objects.

In the knowledge commons, community members create, access, contributes to, and own resources in the commons. In this context, the perspectives of users are not those of traditional users but those of co-producers of knowledge and participants in activities that are meaningful to the communities they belong to. To understand the transformations of cultural institutions our study proposes an approach that is at the intersection of the proposed view of UCD, as CCD, and the knowledge commons.

3 Community-Based Design: A Case-Based Program of Research

In recent times case study research designs have increasingly been argued for their value and ability to investigate contexts within which research questions are raised (Kaplan and Maxwell, 1994; De Vaus, 2001; Yin, 1994). In contrast to experimental design, this study requires and desires little control over parameters as the contexts of information use is the focus of the study. Below we present the case study a farm women's community in Victoria as an example of community centred design.

3.1 The Women on Farms Gathering Community

The **community** and its interactions evolve around annual gatherings held in various locations in the rural areas of Victoria, an Australian State. These gatherings are organised for the purpose of sharing stories amongst farm women, empowering the women in the communities in the process. Although the first gathering was held as early as 1990, the involvement of the Museum of Victoria, as a cultural institution, was not significant until 2001. At the Beechworth gathering that year, objects and the women's stories were brought together as a collection as history boards. With the design and display of these history boards, the community was immediately aware of a collection that is unique to them, and an imminent task at hand. A central, neutral institution that could curate the collection would also ensure the sustainability of it. There was also a larger motivation: this relationship will protect the culture and resources of the community from being lost over time. The Museum was contacted for this purpose.

According to the earlier definition of a museum by ICOM, museums conserve, research, and communicate 'material evidence of people and their environment.' In a narrow sense, this implies physical artefacts that museums collect, hold or exhibit – but museology have also introduced new ways of thinking about these artefacts (Dale, 2003) to include also the stories that are directly related to the physical and material artefacts. This also allows collections to be curated in a participatory sense, having the stories of people to provide diverse and richer accounts of artefacts in the collection. Through these interactions the community and the museum come together to create a knowledge commons that promotes the open sharing of stories, on artefacts that are agreed by the community to be important and symbolic.

Since then the collection grew over time and across a broader geographic area, and the engagement within the community and between the community and the Museum evolved. These deepening interactions highlighted the need for a medium to communicate and exhibit the collection to the community as well as more broadly to members of the public. This led to the formulation of a digital approach to develop the WoFG collection. A detailed analysis of the tasks was conducted leading to the development of a specific approach to define the requirements of the collection. This approach took into account the broader contexts of the partnership between the community and the Museum along with the four layers of interaction (from individual to society).

Because the WoFG was a self-initiated 'grass-root' endeavour, the Gatherings were viewed by the Museum as a particularly valued partnership as they provided unique experiential knowledge, instantiated both in objects and stories. The involvement of the Museum as a cultural institution was intentionally kept as an equal partnership ensuring appropriate engagement with the WoFG community. It is important to note that this effort came from both sides – the institution and the community. Such guiding principles were largely based on participatory action research philosophies that saw the community as a knowledgeable partner, the researchers as collaborators, with a primary goal to contribute to the betterment of the community in its context (Nyden, 1997; McKay and Marshall, 2001).

Researchers and developers involved in the project were engaged in discussion groups and meetings between the Museum and the community. The generation of functional requirements for the information system was therefore largely inspired through observations and first-hand engagement with the partnership between the cultural institution and the community. Thereafter, these functional requirements were developed into technical specifications with developers after considering the various levels and layers of interactions. The information system was designed around the information objects (the stories of women, symbolic icons, banners, oral history recordings, videos, photographs, and other memorabilia relating to the lives of Victorian farm women) and interaction with these object for individuals, in the sharing of stories between individuals (community/group) and for the role of the Museum as a cultural institution charged with the protection of public knowledge within the knowledge commons. This design was also shaped by the broader society, highlighting the duality that exists in these interactions, as indicated by structuration theory.

It is possible that the aggregation of individual needs, when assessed separately, might lead to the same design arrived at when the community's needs are negotiated collectively. However, there would seem to be a better balance of power in the latter approach, effectively transposing the user from an object (the user as a 'victim' to 'rescued') to that of active participant in the design process (an actor). Such an interpretation can be drawn from UCD literature that concentrates on the adeptness of designers and developers, (including usability specialists and managers) to capture the needs of users effectively. Such approaches only provide effective interfaces and systems to 'rescue' end users. This was highlighted by a participant at the beginning of the study who feared IT researchers even if they brought gifts. This attitude might be construed as an unwillingness to be 'rescued' in this way. Lottkowitz (2005) wrote: "I don't mean to caste aspersions on IT researchers as such, but frequently there's a lack of sensitivity about the different needs in rural communities, and ways to get research 'out there' in a credible fashion." She also commented: "… I have a sense from the early discussions about this project that it needs to be women focused and driven, and IT is not always friendly for many women in the communities …"

Yet it is notable that this same participant has championed an e-bulletin for the groups involved in the project. She recognises the irony of her commitment to that system alongside her scepticism about the involvement of IT researchers. This 'us and them' tension is a manifestation of the kind of problem in information systems design practice that our proposed community-based, adaptive approach to UCD seeks to address. Such comments reinforce a view that while there is a lack of fear about using technologies in rural communities; at a deeper level there is a fear and cynicism about the approaches adopted by technology developers and researchers.

4 Conclusion and Future Work

The proposed community-centred design approach was implemented using action research principles in a case study of a women's community and its partnership with a cultural institution. Although the actual development of the information system was completed within six months, the functional and technical specifications were generated through an emergent process using the approach proposed in this paper. Other case studies using the community centred design approach are being conducted that focus on different cultural institutions and different communities to explore other aspects of the framework presented in this paper. On completion of these studies, there is considerable interest to apply this approach to other institutional domains, such as political organisations.

5 Acknowledgements

We would like to also thank Ms Liza Dale-Hallett and members of the Women on Farms Heritage Group for their participation and inputs for this study. The research project has been supported by these Monash University research grants: "Cultivating memory in the Victorian women on farms gathering heritage collection: a digital approach"; "Community building: the power of public library networks"; and "Towards a knowledge commons: modeling a transformation of library services and information resources provision in the vocational education sector".

References

Bailey, R. and Tierney, B. (2002) Information commons redux: concept, evolution, and transcending the tragedy of the commons. *The Journal of Academic Librarianship*, 28 (5), 277-286.

Beagle, D. (1999) Conceptualising an information commons. *The Journal of Academic Librarianship*, 25 (2), 82-89.

Benkler, Y. (2003). The political economy of commons. UPGRADE*: European Journal for the Informatics Professional*, 4(3), pp 6-9.

Bollier, D. (2003). The Rediscovery of the Commons. *UPGRADE: European Journal for the Informatics Professional*, 4(3), pp 10-12.

Bollier, D. (2004). Why we must talk about the information commons. *Law Library Journal*, 96(2), pp 267-282.

Bonner, J. V. H. and Porter, J. M. (2002). Envisioning future needs: From pragmatics to pleasure. In Green, W. S. and Jordan, P. W. (Eds.), *Pleasure with Products: Beyond Usability* (pp 151-158). London: Taylor & Francis.

Burstein, F. and Linger, H. (2003) Supporting Post-Fordist Work Practices: A Knowledge Management Framework for Supporting Knowledge Work. *Information Technology & People Special Issue on Organizational Implications of Knowledge Management Systems*. 16(3). pp 289-305.

Burstein, F. and Linger, H. (2005). "Task-Based Knowledge Management." In Schwartz, D. (ed.) *Encyclopedia of Knowledge Management*. Idea Group Reference.

Card, S. K., Moran, T. P. and Newell, A. (1983) *The psychology of human-computer interaction,* Lawrence Erlbaum, New Jersey.

Carroll, J. M. (2000). *Making use: scenario-based design of human-computer interactions*. Cambridge: MIT Press.

Cockton, G. (2004) From quality in use to value in the world. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)* (pp 1287-1290). Austria: ACM.

Cowgill, A., Beam, J. and Wess, L. (2001) Implementing an information commons in a university library. *The Journal of Academic Librarianship*, 27 (6), 432-439.

Dale, L. 2003 Stories and storytelling: a cultural partnership between Museum Victoria and the Victorian Women on Farms Gathering In *Proceedings of the Setting the Agenda for Rural Women: Research Directions* (70-87). Wagga Wagga: Centre for Rural Social Research.

De Vaus, D. A. (2001) Research design in social research, Sage, London.

DePaula, R. 2003 A new era in human computer interaction: the challenges of technology as a social proxy In *Proceedings of the CLICH 2003 - Building Bridges among Individuals and Communities* (219-222). Rio de Janeiro, Brazil

Fraser, J. (2002) The culture of usability. *New Architect*, Vol. 7, pp. 26-30.

Giddens, A. (1986) *The constitution of society*, Berkeley, University of California Press.

Gulliksen, J., Göransson, B., Boivie, I., Blomkvist, S., Persson, J. and Cajander, A. (2003) Key principles for user-centred systems design. *Behaviour & Information Technology,* 22 (6), 397-409.

Hales, S., Rea, D. and Siegler, M. 2000 Creating a technology desk in an information commons In *Proceedings of the SIGUCCS Conference on User Services* Richmond, Virginia, USA: ACM.

Head, A. J. (1999) Web redemption and the promise of usability. *Online*, 23(6) 20-32.

Hellstrom, T. (2003) Governing the virtual academic commons. *Research Policy*, 32 391-401.

Jokela, T., Iivari, N., Matero, J. and Karukka, M. 2003 The standard of user-centred design and the standard definition of usability: analyzing ISO 13407 against ISO 9241-11 In *Proceedings of the Latin American conference on human-computer interaction* (53-60). Rio de Janeiro, Brazil: ACM Press.

Kaplan, B. and Maxwell, J. A. (1994). Qualitative research methods for evaluating computer information systems In *Evaluating health care information systems: methods and applications*(Eds, Anderson, J. G., Aydin, C. E. and Jay, S. J.) Sage Thousand Oaks, California, pp. 45-68.

Karat, J. and Karat C. M. (2003) The evolution of user-centered focus in the human-computer interaction field. *IBM Systems Journal*, 42 (4) 532-541.

Leadbeater, C. (2006) Libraries and the creative economy. *Symposium: Libraries of the 21st Century*, State Library of Victoria, 23 February 2006.

Linger (2002) *Conceptualising IMS* [diagram]. Unpublished – used with permission.

Lottkowitz, Anna (2005) *Email to Marian Quartly*, 23 May – used with permission.

Lukasik, S. J. (2000). Protecting the global information commons, *Telecommunications Policy*, 24(2000), pp 519-531.

MacWhinnie, L. A. (2003) The information commons: the academic library of the future. *Libraries and the Academy*, 3 (2), 241-257.

Merkel, C., B, Xiao, L., Farooq, U., Ganoe, C. H., Lee, R., Carroll, J. M. and Rosson, M. B. 2004 Participatory design in community computing contexts: tales from the field In *Proceedings of the Participatory design conference* Toronto, Canada: ACM Press.

Moritz, T. D. (2004) Conservation partnerships in the commons? Sharing data and information, experience and knowledge, as the essence of partnerships. *Museum International*, 56 (4), 24-31.

Nielsen, J. (1993) *Usability engineering,* Academic Press, Boston.

Norman, D. (1988) *The psychology of everyday things*. Basic Books Inc: New York.

Norman, D. A. (1986). Cognitive engineering In *User centred systems design* (Eds, Norman, D. A. and Draper, S. W.) Lawrence Erlbaum Associates Inc, Hillsdale, New Jersey.

Norman, D. A. and Draper, S. W. (Eds.) (1986) *User centred systems design,* Lawrence Erlbaum Associates Inc, Hillsdale, New Jersey.

Orlikowski, W. J. and Robey, D. (1991) Information technology and the structuring of organizations. *Information Systems Research*, 2 (2) 143-169.

Pang, N. and Schauder, D. (In press). User-centred design and the culture of knowledge creating communities: a theoretical assessment. In *Proceedings of the 8th Australian Conference on Knowledge Management and Intelligent Decision Support*, 5-6 Dec 2005, Melbourne.

Pang, N., Schauder, D., Quartly, M and Dale-Hallett, L. (2006). User-centred design, e-research, and adaptive capacity in cultural institutions: the case of the Women on Farms Gathering Collection. In *Proceedings of the Asia-Pacific Conference on Library and Information Education and Practice*, 3-6 April 2006, Singapore.

Preece, J. (2000) *Online communities: designing usability, supporting sociability,* John Wiley & Son Inc, New York.

Schauder, D., Johanson, G. and Stillman, L. (2005) Sustaining a community network: the information continuum, e-democracy and the case of VICNET, *Journal of Community Informatics* 1(2) 79-102.

Schneiderman, B. (1992) *Designing the user interface: strategies for effective human-computer interaction (2nd edition),* Addison-Wesley, Reading, Mass.

Schuler, D. and Namioka, A. (Eds.) (1993) *Participatory design: principles and practices,* Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Spinuzzi, C. (2003) *Tracing genres through organizations: a sociocultural approach to information design*. Cambridge: MIT Press.

Taxen, G. 2004 Introducing participatory design in museums In *Proceedings of the 8th Biennal Participatory Design Conference* (204-213). Toronto, Canada.

Yin, R. K. (1994) *Case study research: design and methods,* Sage, Thousand Oaks, California.

ICT Solution and Network Capabilities Development: The Role of the Codification Process in the KMP Experience

Pierre-Jean Barlatier, Catherine Thomas

GREDEG Sophia Antipolis UMR 6227 UNSA-CNRS, France
barlatier@gredeg.cnrs.fr, thomas@gredeg.cnrs.fr

Abstract: The objective of this paper is to explore the contributions of a codification process to the analysis of knowledge creation mechanisms within a network of firms. The idea is to provide a theoretical framework on knowledge creation dynamics within a network through the notion of network capabilities, and to analyse the impact of a codification process on the development and the strengthening of these capabilities. The illustration of this work in progress is located in the scientific park of Sophia Antipolis (Alpes-Maritimes, France), focusing on the Telecom Valley® association which gathers the main actors of the sophipolitan Telecom cluster. Our empirical study is conducted as a part of a research contract, the Knowledge Management Platform (KMP) project. The aim of this contract is to build a semantic web service of competencies in order to enhance exchange and combination dynamics of knowledge within the Telecom cluster thanks to the codification of tacit knowledge. We have centred our analysis on the codification process conducted within the Telecom Valley® network during the conception of the KMP project. The KMP experience shows the main role of the codification process in the enhancement of the organizational knowledge creation mechanisms of the Telecom Valley® network.

1 Introduction

According to Kogut and Zander (1992), and Nahapiet and Ghoshal (1998), organizational knowledge creation is above all a process of social exchange and combination, and absorption of knowledge. It reveals therefore the need to rely more and more on the exchange and the combination of external resources with partnerships and alliances, and conduct firms to open themselves on the outside. Following Håkansson (1993) and Kogut (2000), an organizational form able to offer the benefit of both specialization and variety generation exists: the network. Hence, the network capacities to create, accumulate and transfer collective knowledge are named "network capabilities" (Foss 1999 and Kogut 2000). Based on the previous analysis of Nahapiet and Ghoshal (1998) on social capital, this paper provides theoretical refinements of the concept of network capabilities as well as the role of these capabilities on the knowledge creation dynamics within a network. Thus, we intend to cross network capabilities and codification approaches through the design of an ICT tool, and analyse the impact of a codification process on the enhancement of these network capabilities. In order to do so, we are carrying out a case study located in the scientific park of Sophia Antipolis (Alpes-Maritimes, France). A major association of this park, the Telecom Valley® (TV), aims to build a web service of competencies in order to enhance exchange and combination dynamics of knowledge within the Telecom cluster.

This paper is composed by three different parts:

We identify and analyse in a first part the impact of networks on organizational knowledge creation processes, by studying the different dimensions of network capabilities.

We present in a second part the empirical study of the KMP project within the TV network. We analyse therefore the specific knowledge codification process offered by KMP.

Finally, the third part illustrates the reciprocal influences and interactions between network capabilities, the codification process and knowledge creation dynamics within a Telecom cluster.

2 Knowledge-creation mechanisms within networks

After emphasizing the importance of the social relations in the knowledge creation process, we analyze the network capabilities concept thanks to Nahapiet and Ghoshal's (1998) works.

2.1 Organizational knowledge-creation processes

According to Kogut and Zander (1992), Nahapiet and Ghoshal (1998) and Shawney and Prandelli (2000), organizational knowledge creation is above all a social process. Their work shows that organizational knowledge-creation processes are in accordance with a sociological approach "an emerging, dynamic and diffuse process" where "new knowledge is the output of a synergistic interplay between individual contributions and social interaction" (Shawney and Prandelli 2000: 28). Thus, we can point out that this conception is close to the one advanced by Nahapiet and Ghoshal (1998) who maintain that organizational knowledge creation based on two key mechanisms: exchange and combination.

However, Nahapiet and Ghoshal (1998) have identified four required conditions in order to make exchange and combination as knowledge-creation mechanisms effective: (i) the opportunity to make exchange and / or combination may exist; (ii) the actors must be able to anticipate the value created by the exchange / combination processes; (iii) they must be motivated; and (iv) they must be able to combine knowledge and resources.

Combination and exchange are therefore complex social processes that reflect the interlocking of knowledge forms in an organization able to create and share knowledge, to coordinate, structure, and communicate. The thesis defended by Nahapiet and Ghoshal (1998) is that the social substratum (and the social capital concept in particular) makes the development and the creation of organizational knowledge easier by affecting the necessary conditions in order to enhance the combination and exchange processes. In this perspective, the setting up of a network of firms thanks to spatial and / or technological proximities creates exchange opportunities. A network's efficiency is measured notably with its capacity to transfer information quickly (role of information provider and distributor), but above all in its capacity to lay out various competencies and professions. Beyond information traffic it poses a problem of comprehension (receptivity, adaptability) that can not be solved by a purely technological approach and requires the establishment of a common language.

The network is thus an efficient means for firms for acquiring external resources like knowledge (Håkansson 1993), and these capacities to create, accumulate and transfer collective knowledge are named "network capabilities" (Foss 1999 and Kogut 2000). Hence the idea of combinatorial diversity proposed by Kogut (2000) and Håkansson (1993) lies on the hypothesis of actors' knowledge bases complementarities and network capabilities.

2.2 Network capabilities: what are they?

Network capabilities are seen as beneficial factors that lie outside the individual firm, referring to "…what collectivities of firms – networks – know about the production of goods and services, the organization of this production (network capabilities), and how they learn about it (collective learning)" (Foss 1999: 3); or emerging factors from interactions of individual firm's capabilities and collective learning phenomenon: "network capabilities… are not specific to a firm, but represent joint gains to coordination and learning" (Kogut 2000: 406). According to Foss (1999), network capabilities are accumulated over time and experience.

Thus, network capabilities create value thanks to synergy effects between complementary assets and competencies. The various definitions of this concept proposed in the literature are all referring to coordination and learning gains. Kogut's works (2000) emphasize two dimensions of network capabilities: the structure and the identity. This author highlights then the role of the identity on the generation of rules of coordination.

These works suggest the key role played by network capabilities on the knowledge creation process. However, their definition is still fuzzy and not operational. Since organizational knowledge creation is a fundamentally social process, we suggest thus to explore the social dimensions of networks to refine network capabilities' concept. In order to do so, we propose, as Nahapiet and Ghoshal (1998) have made with the social capital concept, to start with a reflection on the notion of embeddedness developed by Granovetter (1985). Indeed, Granovetter thinks that every social collective (such as a network) is embedded in relational and structural relationships. The relational embeddedness represents the quality of the dyadic relations of the network, the kind of personal relationships that actors have developed between them through the story of their interactions emphasizing then trust relations and cooperative norms. The structural embeddedness represents the general properties of the network as a whole, such as the network configuration and the nature of its ties. Nahapiet and Ghoshal (1998) have added a third dimension, the cognitive one, based on shared representations and meaning systems. Then we propose an enrichment of the concept of network capabilities considering the structural, relational and cognitive dimensions suggested by Nahapiet and Ghoshal (1998), as well as their interactions with the knowledge creation process (Fig.1). This model presents a double interest, first it improves our comprehension of network capabilities, and second, it outlines their role in the creation of organizational knowledge.

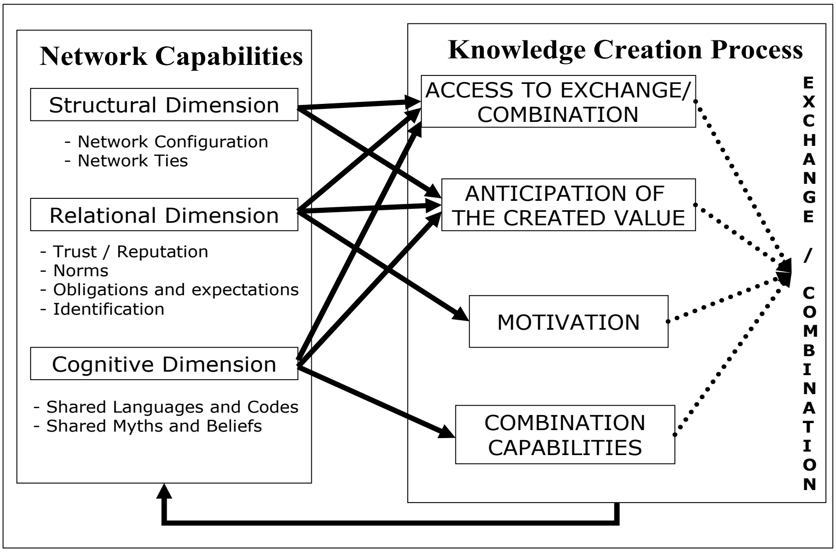


Fig. 1. The Theoretical Model.

We propose to use this model in order to design and implement an ICT solution of knowledge management within a network of firms. Indeed, if ICT enhance actors' connections, they can not substitute for the social substratum inherent to organizational knowledge creation processes. This analysis suggests thus that an ICT solution of knowledge creation should not only enhance information exchanges but has to foster network capabilities by integrating these three dimensions.

3 The KMP Project : the building of a KM solution shared by a community of firms and institutions

The illustration of this study is located in the scientific park of Sophia Antipolis (Alpes-Maritimes, France). Sophia Antipolis, is "one of the most highly publicized technology parks which combine establishments of multinational corporations, small and medium-sized firms, and large public research centers and universities, under the auspices of public regional authorities" (Castells and Hall 1994: 85). We present in a first part our empirical study: the KMP project. In a second part, we detail the uses-based methodology used in the conception process of the KMP solution.

3.1 The Empirical Material: The KMP Project

A non profit business-driven association, Telecom Valley® (TV, founded in 1991 by eight leading actors in telecommunications industry), has initiated in 2002 a project of inter-firm knowledge management, the "Knowledge Management Platform" (KMP). The objective of this project is to propose an innovative Solution including a map of competencies of TV's actors in order to enhance exchange and combination of knowledge within the Sophia Antipolis Telecom cluster. The goal here is to obtain a better identification of actors and projects while facilitating the cooperation and the creation of a shared language between its members. KMP is an experimentation of an ICT infrastructure, a semantic web service of competencies. An abstract representation of competencies based on five points (action, deliverable, business activity, beneficiary and key resources) has been proposed (Rouby and Thomas 2004). These categories constitute an abstract level of codification allowing the creation of codes. These codes ensure the detection of competencies and their comparison depending on actor's interests and vision. Hence these actors can consider appropriate combinations of their choice. These first points for representing competencies are still being used and absorbed by actors. They constitute the first codes shared by the network and the first bricks for building the shared language. This shared language is based on the elaboration of specifics ontologies for each category of codes. An ontology "is an object capturing the expressions of intensions and the theory accounting for the aspects of the reality selected for their relevance in the envisage applications scenarios" (Gandon 2001). Hence, its role is to define worlds constituting the area in which the knowledge will be represented by the actors involved.

An important outcome of the KMP codification process is to convey the nature of the diversity of existing competencies, in other words, their similarity and complementarity, according to Richardson's terminology (1972). Similarity in KMP is related to competencies sharing the same 'action' and 'resources' whereas complementarity is about competencies found within the same 'business activity' that could probably be combined if they belong to different organizations.

The KMP codification process includes various stages: models building, language creation and message writing. These stages are not sequential but iterative, implying the co-evolution of several forms of knowledge (tacit, articulated and codified). The "step by step" approach, developed by the uses-based methodology, allows a progressive implication of actors in order to sustain a process of adoption / adaptation in the prototype building and for generating trust in the codification process.

3.3 A Uses-directed methodology

This study has been conducted through the commitment of researchers in management in the KMP project. Our tasks were to find managerial models in order allow the conception and the implementation of the web service. As we have intervened on and during the research process, we are in the framework of an intervention research (Argyris 1970). Within the framework of our intervention research a high position is granted to the web service's conception which is fully integrated in the researchers-actors interactions process.

The aim of this research revealed an open issue without clearly specified outlines, which has justified the hallmark of the RNRT (French Telecom Research Network) since 2002 to 2005, as an exploratory project. Hence, a multidisciplinary approach was essential to the understanding of Knowledge Management practices. Consequently, the KMP project involved researchers from socio-economic sciences (GREDEG), cognitive sciences (INRIA), telecommunications (ENST) and practitioners and users (TV) for a total force of 187 men / month for a three years period. The conception of the solution is based on "uses scenarios" which occur at the same time upstream from the tool for its conception and downstream for its evaluation. Uses scenarios describe the inter-organizations recurrent interactions models which define in observable and behaviouristic terms the essence of actor's roles and their communication strategies about their competencies.

In addition, uses-based methodology implies a co-conception of the tool and a co-evolution of conception and uses. An iteration of conception/experimentation loops is then necessary, given that every loop is the occasion of enrichment to each step. The project is at the moment in its fourth loop, the first one took place in 2001-2002, during its conception.

4 Results and Discussion

The analysis reveals two distinctive categories of results, one considering the impact of the KMP solution on the different dimensions of network capabilities and another one revealing the enrichment of these mutual influences by highlighting the dynamic interactions between network capabilities' dimensions, codification processes and knowledge creation conditions.

4.1 Impact of the development of network capabilities dimensions on knowledge creation conditions

Given the recent (and still limited) implementation of the KMP prototype, we are able to identify four main kinds of ties that illustrate the relations between the impacts of the codification process on the several network capabilities' dimensions and their consequences on knowledge creation conditions.

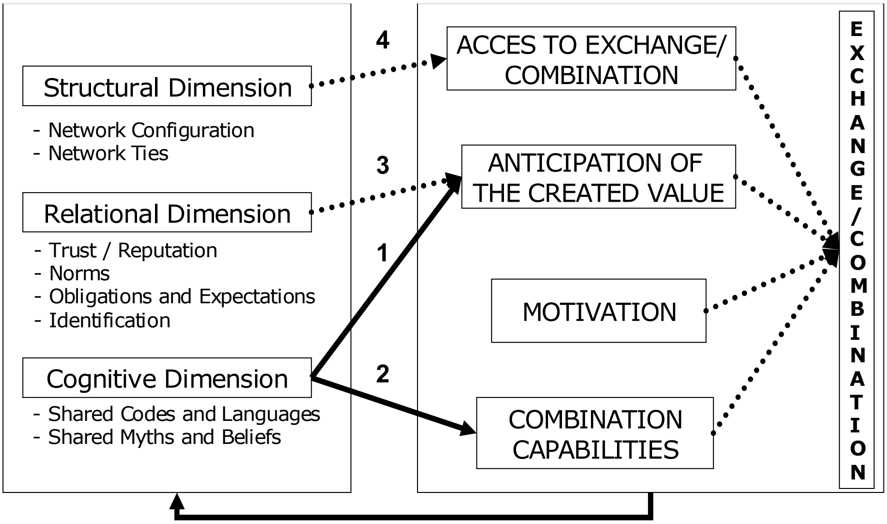


Fig. 2. KMP's impact on network capabilities and knowledge-creation conditions.

The ties 1 and 2 (Fig. 2) emphasize the influence of the cognitive dimension on the anticipation of the created value by cooperation and the combination capabilities. The main effect of KMP comes from the creation of shared languages and codes such as ontologies of competencies. Indeed, we have noticed, that the representation of the "sophipolitan territory" allowed by the use of complementarity / similarity notions, has made possible the identification of competencies poles (similar competencies) and value chains (complementary competencies). This identification was largely appropriated by network actors, leading therefore to two main results:

- Every club and association that aimed to enhance synergies within the whole sophipolitan cluster has been mapped. Indeed, some were classified as aiming to foster complementarities (more market oriented) and others as aiming to foster similarities (more oriented towards technological innovation). This was presented during the "Club's Day" event by members of KMP's steering committee and has solved many problems of roles and legitimacy of clubs and associations. Hence, the diffusion of KMP methodology beyond the telecom cluster shows the important needs in terms of representation and of development of combinative capabilities;

- Moreover, this mapping methodology based on the "similarity / complementarity" concepts have also been adopted and reused by the PACA region in order to present a national project about competitiveness poles, the "SCS" pole (Secured Communicating Solutions). Hence, these concepts have really structured the presentation of cooperation projects within the pole, showing then their impact on network's combinative capabilities and anticipation of created value. This ability to manage these concepts appears to be a source of competitive advantage, as in this specific high velocity industry, increasing technological knowledge requires the capability to reduce mismatches between the production and demand sides (Antonelli et al. 2001).

Thus, we have observed KMP's various effects on the network's cognitive dimension, providing notably shared languages and codes such as ontologies and common space representations that develop actor's abilities to anticipate created value (tie 1 in Fig. 2). Hence, these specific communication codes represent, according to Kogut and Zander (1992), an asset source of value creation. In addition, we can also quote another significant effect, regarding the software engineering and consulting "pilot" firms involved in the KMP process of codification. Indeed, during the process of construction of ontologies in which they have played an active role, these pilot firms have realized that they can gain more in being partners than being fierce competitors. Most of them have today the project to develop partnerships about joints solutions, aiming to reach more and bigger customers, within and outside Sophia Antipolis. They even think about creating a new association, composed exclusively with software engineering and consulting firms. Therefore, we have effectively here another concrete example of the impact of KMP on the cognitive dimension, and its consequences on conditions of knowledge creation such as anticipation (the will to cooperate, tie 1 in Fig. 2) and combination capabilities (creation of a new association and mutual projects, tie 2 in Fig. 2).

The third tie (Fig. 2) shows the main impact of KMP on the relational dimension. This comes from the process of codification of obligations and expectations as well as norms of cooperation, taken from the identification of "best practices" in matter of cooperation, revealed by executives of biggest firms (i.e. biggest customers) of the network. Indeed, during this codification process, we have seen that largest companies were looking for partners and / or subcontractors that present not only required technological competencies but also managerial and organizational competencies such as project-team management, total quality management, etc.; another request was to include several "levels of accessibility" about information, in order to have the possibility to give more detailed information to privileged partners if one needs to. Expectations about these managerial skills allow them to better anticipate the value created (tie 3 in Fig. 2). TV has always been concerned about membership rules, and some have progressively emerged, before KMP, through various prizes such as the CLIPSAT Trophy (rewarding the best ranked sub-contractor of TV) or the "Innovation Prize" (rewarding the most original project in the Telecommunications domain or related services). The codification of these rules of cooperation has quickly appeared as a *sine qua non* condition to the realisation of KMP. They are capital in the success of an inter-firm knowledge management solution which aims to develop partnership dynamics. TV has even modified its CLIPSAT Trophy, adding requirements about the accuracy, the authenticity and the pertinence of the companies' description and updating of organizational competencies in the KMP base. Through KMP, TV has implemented strong membership rules, in trying to elaborate a professional code of ethics and subcontracting, which enhances trust relations and reputation effects that have a positive impact on anticipation of value through exchanges (tie 3 in Fig. 2). KMP has also developed the identification process within the network thanks to managerial models displayed during the codification process. This shared representation allowed the sharing of "similarity" and "complementarity" concepts taken from Richardson's (1972) works on cooperation. In this case, "similarity" represents competencies sharing the same resources and the same actions; "complementarity" represents competencies aiming to a same supply system. The common representation and the introduction of these concepts enhanced actor's capabilities to anticipate the created value through partnerships, as they were able to select the best cooperation opportunities and perspectives (tie 3 in Fig. 2).

The fourth tie (Fig. 2) shows the main impact of KMP on the structural dimension. Indeed, the use of KMP makes easier the identification of new contacts for network actors as well as external ones, and makes these potential partners also easily reachable. We quote, for instance, a user from a company [X], which has experienced the KMP solution and has voluntarily shared his experience with us in sending the following e-mail: "*I have a selective request: 'company [X]' is looking for on-board applications editors/integrators on symbian tel. or java environment which are located in the 'Alpes-Maritimes' department. I have found in the KMP base 'company [Y]' in IT Services, Solutions and Applications Providers. Do you know about other companies?*" As a consequence, the KMP web service, in allowing the multiplication of inter-organizational links, develops exchange and combination opportunities within the network, and *de facto*, intensifies the dynamics of partnership (tie 4 in Fig. 2). We can however notice that these new ties are initially weak.

Globally, we can observe from these results that they confirm in a way Nahapiet and Ghoshal's perspectives about the effects of the different dimensions on knowledge creation conditions. But as a second kind of results, our empirical experience allows us the detection of multidimensional effects on organizational knowledge creation conditions.

4.2 The enrichment of multidimensional, dynamic interactions

We have seen that the implementation of the KMP web service of competencies has developed network capabilities dimensions and knowledge creation conditions. But beyond the "simple" dynamics we have shown, the KMP codification process and the construction of managerial models have generated complex, multidimensional dynamics that have affected at a same time network capabilities, organizational knowledge creation conditions and the coherence of the TV network.

Indeed, the Telecom value chain has provided a collective representation to every TV's actor that had as main consequence a collective realization of the network boundaries, mass effects and actor's games of interests. This awareness of TV's current situation by its own actors has rapidly triggered off a dynamic of change, because they have quickly detected the network's shortcomings for innovation and have found reasons of their collective inertia. This gave then indications about the current level of coherence of the Telecom cluster, which had an impact as well on demand and profile choice of newcomers and on trajectories of members' strategic diversification, i.e. their games of interests. For instance, we can notice the recent request of a small network of firms (Multimed) working in multimedia and content, located outside Sophia Antipolis but in the PACA region, which wants to join TV after having heard about the KMP experience. A few years ago, they wouldn't have applied for a TV membership, as they are geographically and technologically quite far from TV members, lacking information and anticipation abilities. Today, thanks to the Telecom value chain model, this potential newcomer is able to detect opportunities *via* the KMP showcase, and TV has now increased its possibilities of selecting the right newcomers, of widening the network's knowledge base, of determining their specialization trajectories, etc.

Hence, this representation provided by the codification process and the collective self-consciousness about network shortcomings has had a great impact on both cognitive and relational dimensions of the network, i.e. its identity. This collective comprehension of TV's identity has provoked an "organizational closure" of the network, because TV's actors were, at this point, able to define precisely the boundaries of the network, and consequently, its knowledge base. This closure is then paradoxically the source for appropriates perspectives of network's openness, and is today a key element of the network's coherence management.

Consequently, this led to a significant impact on both relational and cognitive dimension of network capabilities, creating thus a genuine collective "savoir-voir" (cognitive dimension) that influences mainly access and motivation conditions, as showed in Figure 3:

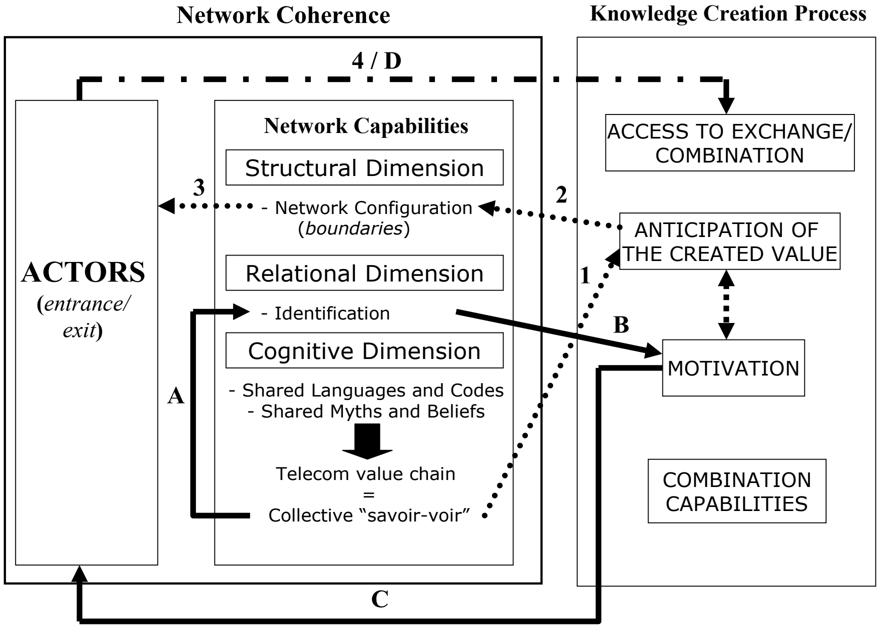


Fig. 3. Multidimensional interactions and knowledge creation conditions.

In this Figure, we can see that the collective "savoir-voir" resulting from the cognitive appropriation of the value chain model by the actors leads to an example of complex dynamics:

- Indeed, by mapping current exchanges, the value chain revealed profitable future opportunities that increased anticipation abilities (tie 1 in Fig. 3), leading to the necessity to open network frontiers and to alter then its configuration (tie 2 in Fig. 3) in order to let new external actors enter (tie 3 in Fig. 3);
- Simultaneously, the value chain model, by determining network boundaries, has developed identification abilities of network actors, that can henceforth benefit from a clear representation of themselves within the network (tie A in Fig. 3), enhancing consequently their own motivation to develop exchanges in order to influence, to master their position within the network (tie B in Fig. 3). This can lead also to the entrance of new actors or the exit of current ones (tie C in Fig. 3).

Then, we have seen two different dynamics engendered by the codification process that have impacted network capabilities dimensions, leading simultaneously (or not) to the necessity of altering its coherence (by the entrance of new actors or exist of current ones) in order to develop negotiated opportunities, i.e. access to exchange and combination (ties 4 / D in Fig. 3), primordial condition for knowledge creation (in particular in a long time perspective). Thus, this codification provides TV a major opportunity to manage its own dialectic heterogeneity /

specialization, i.e. its coherence. We can also highlight that codification processes have reduced cognitive dissonances within TV actors, i.e. enhanced their cognitive proximity, which *in fine* has allowed TV to develop its coherence, while widening its geographical proximity (to the PACA region) and its technological proximity (multimedia and content).

5 Conclusion

In this paper, we have proposed an enrichment of the concept of network capabilities as well as a reflection on their role on organizational knowledge creation, based on Nahapiet and Ghoshal's theoretical works (1998) about social capital. Then we have shown, through an empirical study based on the design and implementation of a web service of competencies, the role of the codification process on the strengthening of network capabilities.

This study has allowed the expanding and the generalization of theoretical propositions in Yin's sense (1989): indeed, this experimentation allowed the refining of simple relations between social dimensions and knowledge creation conditions theoretically identified by Nahapiet and Ghoshal (1998), with partly validating them empirically. Nevertheless, we have outlined the existence of complex, multidimensional dynamics, where the codification process has appeared as being predominant. Hence, we have noticed that the representation of a network's common space through codification processes engendered by the creation of an ICT solution have had a massive impact on both relational and cognitive network's dimensions. As a result, a collective "savoir-voir" has emerged and has definitely developed the network's identity, provoking then an "organizational closure" that has paradoxically given to the network the possibility to open itself from the outside without harming its coherence. This opening has consequently transformed the network's structure by the attraction / searching of new actors and the alteration of the nature of its ties.

References

Antonelli C., Gaffard J.L., Quéré M., 2001. "Interactive learning and technological knowledge: the localized character of innovation processes", "Nouvelle économie" Conference. Sceaux (France), May 17-18.

Argyris C., 1970. Intervention Theory and Method, A Behavioral Science View. Reading MA, Addison Wesley.

Barlatier P.-J., 2003. "Knowledge creation within a localized network of firms in the Telecom industry: the role of communities", XIXth EGOS Conference. Copenhagen (Denmark), Jul. 3-5.

Castells M., Hall P., 1994. Technopoles of the World, the making of 21st Century Industrial Complexes. Routledge London.

Foss N.J., 1999. Networks, Capabilities, and Competitive advantage. Scandinavian Journal of Management, Vol. 15. 1-15.
Gandon F., 2001. “Engineering an Ontology for a Multi-agents Corporate Memory System.” INRIA Research Report, INRIA, France -, http://www.inria.fr/rrrt/rr-4396.html
Granovetter M., 1985. Economic Action and Social Structure: the Problem of Embeddedness, American Journal of Sociology, n°91: 481-510.
Håkansson H., 1993. Networks as a mechanism to develop resources, In Beije P., Groenewegen J., Nuys O. 1993, Networking in Dutch Industries, Garant, 207-223.
Kogut, B., 2000. The Network as Knowledge: Generative Rules and the Emergence of Structure. Strategic Management Journal, 21: 405-425.
Kogut, B., Zander, U. 1992. Knowledge of the firm, combinative capabilities, and the replication of technology. Organization Science, 383-97.
Nahapiet J., Ghoshal S., 1998. “Social Capital, Intellectual Capital, and the Organizational Advantage”, Academy of Management Review, Vol. 23, 242-266.
Richardson G.B., 1972. The Organisation of Industry. The Economic Journal, September: 883-896.
Rouby E., Thomas C., 2004. La codification des compétences organisationnelles : l’épreuve des faits, Revue française de Gestion, mars/avril, n°149, 51-68.
Shawney M., Prandelli E., 2000. Communities of Creation: Managing Distributed Innovation In Turbulent Markets. California Management Review, Vol. 42, N°4: 24-54.
Yin R.K., 1989. Case Study Research: Design and Methods, London, Sage.

Ontology as an Information Base for a Family of Domain Oriented Portal Solutions

Michal Barla, Peter Bartalos, Peter Sivák, Kristián Szobi, Michal Tvarožek, Roman Filkorn

Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies,
Slovak university of Technology in Bratislava, Slovakia.
{barla,bartalos,tvarozek,filkorn}@fiit.stuba.sk, szobi@chello.sk, psivak@mail.t-com.sk

Abstract: Ontologies are becoming increasingly accepted as a form for information representation of distributed web-based information and its processing. In our paper we build a web-based application around an ontology, which serves both as a data base and metadata source used in processing of the data. We propose techniques and methods that exploit this metadata to provide an easy and flexible implementation of the CRUD pattern. We identify patterns in the ontological representation of domain entities and transform them into web-based forms for data management. Based on this flexible framework we can model variations to adapt the application for specific sub-domains.

1 Introduction

A significant amount of resources has already been invested into the development of web-based applications that use ontologies for data storage [6]. Among other advantages, ontologies support the execution of semantic queries on an ontological database, explicit and inherent access to metadata and reasoning based on the formal representation of both the data and metadata.

By definition, an ontology is an explicit formal specification of a shared conceptualization (of a domain) [4]. Ontological languages provide means for the basic modeling of concepts such as classes, relations and various properties. Based on the corresponding language constructs, domain specific concepts can be defined and their semantics specified. Consequently, the model of a domain consists of both concepts defined by ontological languages, which are reused in other domain models and domain-specific concepts unique to a particular domain or set of domains.

If we create a model of a specific domain represented by an ontology, this model will necessarily have to be changed at some time in the future to compensate for changing requirements. Ontologies are especially suited for such change and provide flexible means of data storage.

However, the flexible data storage provided by ontologies would be of limited use unless applications are flexible enough to accommodate a similar degree of change. Otherwise, for any change in the ontology, the corresponding application code would have to be manually updated, which is unpractical for real-world applications.

Thus the apparent flexibility of ontologies results in the need for flexible applications. If we consider forms as the main mean of communication between portal and its users we come to the need for flexible form generation tools. These would be based not only on data stored in ontologies but also on additional (meta)data, which would be needed because domain ontologies shall not contain the information about the desired visual data representation, preserving their generality and delegating application-specific information to other sources.

2 Proposed approach

In our approach, we take advantage of the native availability of metadata in ontologies, which make the data self-descriptive and allows for effective searching in the stored data. Based on the assumption that an ontology may represent a formal model of an information domain, we propose the use of such metadata to build a domain oriented web portal solution for a particular domain.

In order to process ontologically structured data by the state of the art GUI frameworks, we implement mapping tools that transform data between its graph representation and an object-oriented representation. Consequently, we define the mapping of modeling concepts between these two modeling paradigms.

To design a flexible framework, we identify patterns as repeating structures (sets of concepts and their relations) in ontological representations and define consecutive data and processing around these patterns. We assume that similar patterns are shared between ontologies with the patterns themselves being defined using different types of ontological concepts - classes, relations between them, properties, instances and restrictions.

We map identified patterns onto sets of visual elements (graphical user iterface widgets) that correspond to the data stored in a specific ontology. Based on this approach we implement the CRUD pattern (Create-Retrieve-Update-Delete) for a particular entity in a domain ontology.

Since the metadata available in ontologies are not always sufficient to fully create a satisfactory user interface, we define additional metadata, which describe the arrangement of visual components on web-based forms.

Finally, we automatically generate the forms corresponding to ontological concepts, where first the respective ontological patterns are identified. Next, the proper graphical representation is determined and lastly the form descriptions are saved and used during operation.

3 Object--ontology mapping

The creation of an object-oriented representation of ontological concepts introduces new challenges due to the fundamental differences of both representations. These come from the differences between description logic and object-oriented systems and lie primarily in the completeness and satisfiability. Ontologies have a significantly higher expressivity compared to object-oriented approaches [3].

We automatically generate a set of Java bean classes, each corresponding to an integral part of an entity described by the ontology. We represent literals by a simple data type field of the appropriate type and each object property by a separate Java bean.

In order to process these Java beans and store the values in an ontological repository, we need to generate additional metadata for the mapping between objects and classes and properties (RDF graphs) of the ontology. These metadata allow us to bind the ontological class to Java bean fields and include the name of the Java bean, the names of its fields, information about the corresponding OWL properties (e.g. multiplicity, data type) and the type of the object in the RDF triple (object type or a data type - literal).

The mapping itself is performed by a pair of graph-to-bean and bean-to-graph transformers (analogical to O/R mapping in relational databases [2]). These transformers work the graph representation of RDF and use reflection to invoke *get* and *set* methods on the generated Java bean objects.

The transformation process is performed recursively, in each step performs mapping between one object and an integral part of the RDF graph, i.e. object property. Finally, a simple Java bean field is mapped to an RDF triple and vice versa, with the name of the field being used to determine the corresponding ontological property. Thus a simple data type field corresponds to a literal in the ontology and an object property corresponds to an instance of the respective Java bean class.

4 Pattern types

We identified two distinct types of patterns that can be applied at different levels of abstraction and are thus useful for a broad set of ontologies:

- *Widget patterns*, which are based on basic ontological language concepts and correspond to relatively simple configurations in ontologies.

- *Visual patterns*, which define the higher-level visual style of forms, the layout of individual widgets and other form controls (close to user interface design).

To achieve independence from the used ontology we base our metadata processing on basic ontological language concepts. Moreover, we specify a model for the representation of variability in information sub-domains, e.g. specializing the structure and behavior of an instance of the CRUD pattern of a domain entity for some specified subset of users.

4.1 Widget patterns

Widget patterns focus on the structure of classes, subclasses, properties and possible restrictions. They determine the widgets used in a form to edit instances of classes.

4.2 Simple widget patterns

Primitive datatype properties

Patterns for *primitive datatype* properties include all properties, whose ranges are literals (string, integer, float, date, boolean etc.). The graphical representation of these patterns is straightforward - the *rdfs:label* of each property is displayed next to the input field for its value. This would generally be a dropdown list for Boolean values (true, false, undefined), a calendar for date values and a *textbox* for text strings. It is more convenient to use *textarea* input field for longer strings as it increases the readability of the form. To distinguish cases where a normal *textbox* and where a *textarea* should be used, we can either define additional metadata about desired form template or we can compare the average length of existing instances to a predefined threshold.

If the respective property has multiple cardinality, the above elements would be placed inside a special control called *repeater*, which enables users to add/remove more values by means of additional buttons for these actions.

Same-range object properties

Patterns for *same-range object properties* identify classes that have several object properties with multiple cardinality and the same range (class or a union of classes). In this case, the fields for the range are displayed only once and an additional component is used to distinguish the specific property that is being edited. Such a component can be either a dropdown list or a set of radio buttons. All these components are wrapped in a repeater to allow multiple values to be added.

For example, the *Prerequisite* class form the Job Offer domain ontology of the NAZOU [5] project. Each prerequisite has two multiple properties: the *Requires* and *Prefers* which have as their range a union of experience and qualification classifications. Thus, the visual representation provides a radio button to select between the *requires* and *prefers* properties while the rest of the widget is common for both properties and is used to assign experience and qualification prerequisites for job candidates.

Enumerations

Patterns for *enumerations* identify object properties, whose range is a class fully defined by its instances. It should not be possible to add new or edit existing instances. Enumerated classes are defined in the ontology itself, e.g. a class representing the days of the week or various time periods (hour, day, month etc.).

Several graphical representations of this pattern exist. If the cardinality of an object property whose range is an enumerated class is single then it can be represented by a dropdown list of its instances. If the cardinality is multiple, the mentioned dropdown list can be wrapped in a repeater or instances can be represented by a multi-choice *listbox*.

Besides enumerated classes, also classes which allow users to create new instances or choose an existing one can be identified. This information must be stored in the metadata for the appropriate class. In this case all fields necessary to create an instance would be displayed and a *dropdown list* or *listbox* with existing instances would be added as mentioned above.

4.3 Tree hierarchies

Since hierarchies offer a wide range of values, they must be structured in a way that allows users to easily understand and choose amongst them. For example, when users want to choose a country where a company is based, it might be convenient for them to first choose a continent, then a country on that continent, etc.

In ontology, there are two basic ways to represent tree hierarchies. The standard property *rdfs:subclassOf* between classes which represents an *is a* relation and/or custom defined properties between instances can be used, which define relationships between nodes in the tree hierarchy.

One can assume that if a class in an ontology has a property which is transitive and points to instances of the same class (its range and domain are the same), then it is used to represent some form of hierarchy. The job offer ontology of the NAZOU project [5] defined two properties in this way: the *isPartOf* property and the *consistsOf* property, which were mutually inverse and allowed for navigation in a hierarchy of regions.

Additional characteristic of a class is the number of levels of its subclasses and also whether or not a class is fully defined by its subclasses. A class is fully defined by its (direct) subclasses if every individual belonging to that class must belong to at least one of its (direct) subclasses.

If a class is fully defined by its subclasses and only has direct subclasses, we represent it by a dropdown list component that contains labels of these subclasses. Figure 1 and 2 shows the example of the *jo:Benefit* class. Its graphical representation is a simple dropdown list and since *jo:Benefit* is a multiple object property of another class (*jo:JobOffer*), it is wrapped in a repeater (bottom).

In this way, users choose the type of instance they want to create. If the class is not fully defined, the dropdown list will also contain the label of the parent class to enable users to create an instance of it instead of its subclasses.

If there is more than one level of subclasses, their presentation should enable users to browse them. If there are only a small number of classes, it is suitable to display them in a dropdown list as in the previous case and indicate the hierarchy by adding a symbol before the actual name of each class (for instance one dot for each level of hierarchy). This approach might not be suitable to display complex and deep hierarchies, where it is better to create a component which simulates the navigation in a tree. E.g. a *listbox*, which contains all classes of the same level and is redrawn with the classes of the next level when the user chooses one value.

The current location in the tree would be indicated next to the component and users would have the ability to return to a higher level in the hierarchy (e.g. a button, hyperlinks in the path).

The same approach can be applied to a hierarchy created by transitive properties between instances. Whether the user can choose a class or instance which is not a leaf of a tree hierarchy is determined from additional metadata about the class. Metadata are also used when both classes and instances are used to define a hierarchy.

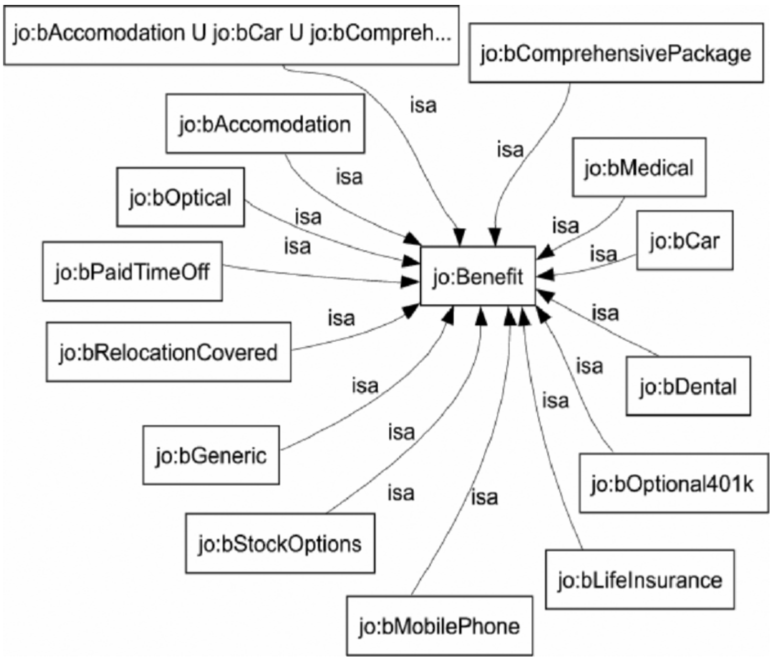


Fig. 1. Example of class *jo:Benefit* that is fully defined by its direct subclasses.

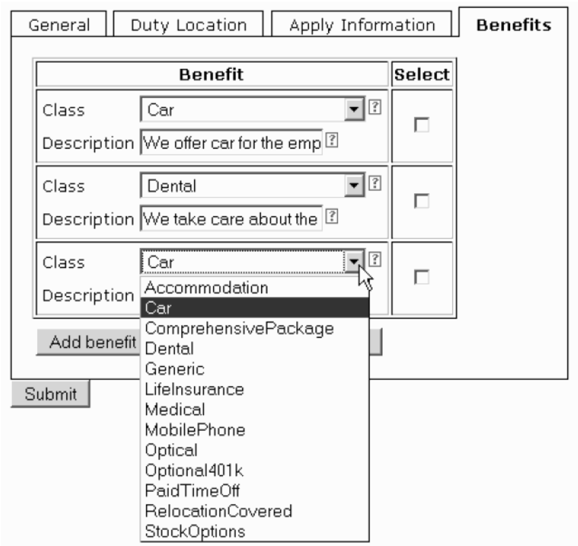


Fig. 2. Visual representation of benefits.

4.4 Visual patterns

Visual patterns describe the concepts of sub forms and identify the typical structure, when a class has an object property pointing to another class (which can also have object properties). So, if class A has an object property pointing to class B, there are several possibilities how to represent it in a form for class A:

- Dedicating an area of the form for the properties of class B, usually bounded by a rectangle. This representation is suitable when class B has only few datatype properties.
- Dedicating a special part of the screen to display any object property of class A and creating a button for each object property of class B. Users choose the property, they want to edit by clicking the appropriate button. If a user clicks on a button of a property in class B, the content of the dedicated part would be redrawn and would contain the form for the editing of the instance of class B.
- Creating a button for each object property of class A. If a user clicks a button for class B, a pop-up window would be shown, where the object property could be edited. This solution is not desirable since pop-ups are usually annoying and are often filtered by web browsers.
- Using a tabbed interface, where each tab would represent one object property of class A. Selecting a tab for class B makes it editable. If class B also contained other object properties, a second row of tabs would be available.

5 Form Generation: Mapping patterns to form controls

The basic principle of dynamic form generation lies in the use of data stored in the ontology along with the appropriate metadata to identify ontological patterns. Patterns are used to define a binding between the data in an ontology and their graphical representation to create forms for specific classes and form controls for specific properties.

Since one pattern can have more than one graphical representation, additional metadata must be used to choose the most appropriate one.

Form generation is a recursive process which begins from the given identifier of a class from an ontology for which the form should be generated and continues via its object-type properties. During form generation, the visual description and the data model of the form must be generated. Furthermore, the corresponding implementation related objects such as Java classes (e.g., Java beans), which store form data must also be generated as well as mapping rules between these classes and the respective ontology. These mapping rules are used by the previously mentioned graph-to-bean and bean-to-graph transformers.

The proposed method matches the structure of each class to patterns described in the previous section. This determines whether the process of form generation is recursively applied to object properties of the class or is terminated by defining a set of simple widgets for display. The matching itself is done by examining the conformance of a selected class to a sequence of patterns in predefined order.

5.1 Form layout

The previously described *widget patterns* map the structure of an ontology to the form controls but do not describe the layout of the form. Although we already discussed the visual representation of object properties between classes (*Visual patterns*), none of the proposed solutions (e.g., tabbed interface) provide information about ordering of the visualization. Moreover, these representations do not allow the personalization of forms preventing us from creating various specialized form templates which fit the needs of individual users. These include hiding of unnecessary form elements or pre-filling forms with default data.

Another problem is that ontologies normally do not contain application-specific information, e.g. whether users are only allowed to choose from existing instances or have to create new ones. Finally, ontologies may contain concepts that should not be displayed on specific forms at all.

Since the lack of flexible form layout support would degrade the proposed solution we defined an additional ontology (fig. 3) which contains information about the order of form tabs and their titles as well as the order of the class properties displayed in these tabs. Additional metadata define whether users are allowed to create new instances of certain classes or are only allowed to choose from existing ones, or the combination of both. The ontology can also describe predefined values for specific fields thus allowing for the creation of forms which are more user-friendly compared to forms generated by generic ontology editors such as Protégé (http://protege.stanford.edu).

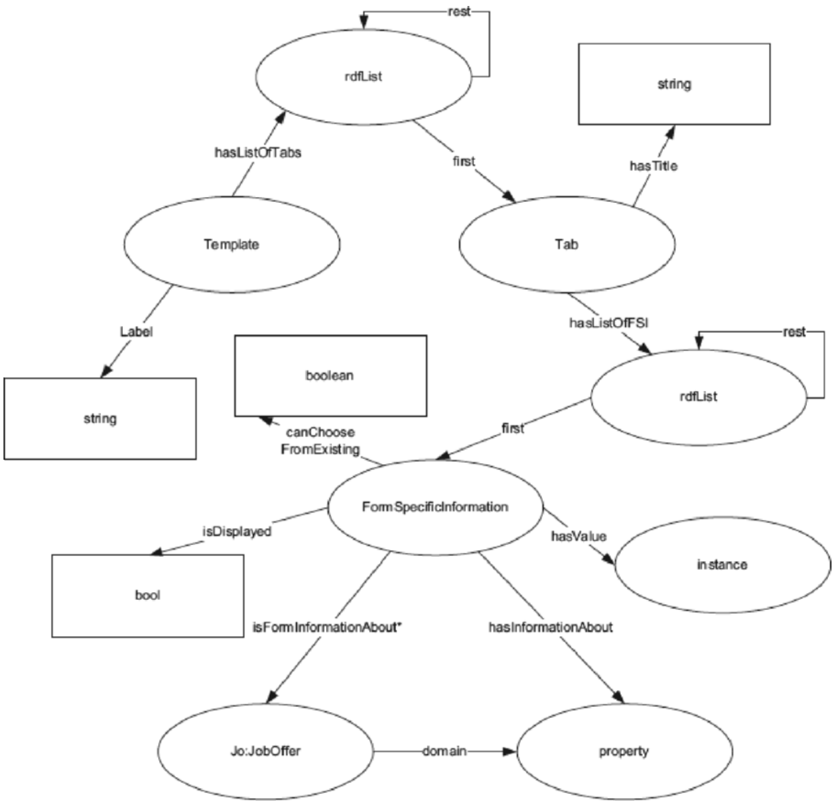


Fig. 3. Meta-model of a form layout information.

The model in fig. 3 shows that our solution supports the reuse of its parts since one Tab instance can be used in multiple lists and one *FormSpecificInformation* instance can be used in multiple tabs. This allows for quick customization of a form template, where some parts of a form can be reused and new layout can be defined for the rest of the form.

The fact that the model is directly connected to a domain ontology (class jo:JobOffer in our case) allows for easy personalization of a form template to meet the needs of organizations and individuals. If a system determines that a user always fills in the same value in a field (e.g., duty location or qualification prerequisites) it can create a template for this user which does have these fields pre-filled with the appropriate values (instances in the ontology). On the other hand, if the user is constantly ignoring some field of the template (e.g., level of management or salary bonus) the system can hide these fields from the template while still allowing their use if user explicitly requests the complete form.

The identification of the template that is used for a specific user is stored in a user model which is used for adaptation throughout the whole system if we consider the CRUD pattern as a part of a larger system.

6 Conclusions

We briefly described one of the current trends – the drive towards web applications built around the semantic web principles and technologies. We also explored the problems introduced by the dependency of portal systems on ontologies as means for information storage.

We proposed dynamic form generation as a possible approach which would accommodate changes in ontologies by increasing the flexibility of web portals. We identified and described several patterns in ontologies and their mapping to form controls. With this knowledge we designed an algorithm that dynamically generates form descriptions from ontologies, which can be used by portal solutions to display and process forms for instances from these ontologies. We find this approach suitable for easy creation of flexible forms for portal solutions with similar tasks. Typical usage is in the domain of offers (job offers, realty offers, vacation offers etc.) and in solutions where CRUD pattern is to be realized by system's users.

To verify the feasibility of proposed solution, we created a job offer portal that uses the job offer domain ontology developed as a part of the NAZOU project [5]. The portal enables users to input and publish job offers by filling in various aspects of the respective job offers in appropriately generated web-based forms.

Future work might include the identification of more complex patterns in ontologies and their mapping to form controls. Exploring the possibilities offered by adaptive hypermedia technologies and their relevance to dynamic form generation is another promising direction of research.

Acknowledgement

This work was partially supported by the State programme of research and development "Establishing of Information Society" under the contract No. 1025/04 and by the Scientific Grant Agency of Slovak Republic, grant No. VG1/3102/06.

References

1. Aldred L, Dumas M, Heravizadeh M, Hofstede A (2002) Ontology Markup for Web Forms Generation. In: Workshop on Real World RDF and Semantic Web Applications

2. Ambler S W (2006) Mapping Objects to Relational Databases: O/R Mapping In Detail. http://www.agiledata.org/essays/mappingObjects.html
3. Battle S, Jiménez D, Kalyanpur A, Padget J (2004) Automatic Mapping of OWL Ontologies into Java. In: Frank Maurer and GÄunther Ruhe (eds) Proc. of the Sixteenth International Conference on Software Engineering & Knowledge Engineering (SEKE'2004)
4. Benjamins V R, Fensel D, Studer R (1998) Knowledge engineering : Principles and methods. Data and Knowledge Engineering, 25(1-2): 161-197
5. Bieliková M, Návrat P, Rozinajová V (2005) Methods and Tools for Acquiring and Presenting Information and Knowledge in the Web. In: Proc. of International Conference on Computer Systems and Technologies CompSysTech' 2005. Varna, Bulgaria
6. Dong J S (2004) Software modeling techniques and the semantic Web In: Proc. of 26th International Conference on Software Engineering (ICSE'04), pp. 724-725, IEEE
7. Obrst L (2003) Ontologies for semantically interoperable systems. In: Proceedings of the twelfth international conference on Information and knowledge management (CIKM '03). CM Press 366{369, New Orleans, LA, USA

Business Rules in Clinical Trials

Olegas Vasilecas, Evaldas Lebedys

Vilnius Gediminas Technical University, Vilnius, Lithuania,
olegas@isl.vtu.lt, evaldas@isl.vtu.lt

Abstract: The paper focuses on the scope of the clinical trials. As clinical trials are rigorously controlled, it is important to have a well-designed control mechanism for clinical trials quality assurance. Each clinical trial is executed according to a set of rules. All the rules for a particular clinical trial are described in different source documents and are not stated explicitly. While rules for particular trial are not organized as a whole, it complicates the control of the trial. The need for accurate recording and processing of patient data is fundamental to any clinical trial. If data stored in the clinical trial database are incorrect, conclusions of the analyses will also inevitably be incorrect. The need of a central storage for rules in particular trial is obvious. The clinical trial business rules repository may be used to reuse the knowledge gathered in one clinical for further trials. The paper presents a method for business rules retrieval from UML models and the employment of proposed in conduct of clinical trials. The analysis of business rules approach principles application in clinical trials is presented.

1 Introduction

Clinical trials are people based studies aimed on new drugs or treatment procedures testing. Clinical trials are conducted to find out whether the new drug or treatment is safe and effective. Clinical trials are performed in all areas of medicine. All clinical trial procedures are documented in detail, because conduct of clinical trial is supervised by local and international authorities. All steps of a particular clinical trial are executed in accordance to specific explicitly expressed rules. Management of these rules is essential in each trial, because it influences the quality of the trial. The appliance of business rules approach in clinical trials is already the subject of researches [4], but the job already done is focused on the clinical trial protocol design improvement. Business rules in clinical trial applications cannot be used to do any changes in the data automatically [7]. Business rules affecting a particular trail are from these types of rule sources:

- clinical trial related documents – clinical trial protocol, clinical trial data validation procedure description, special sponsors requirements, etc.;
- documents representing generic requirements for clinical trials – standard operating procedures for clinical trial design and management, Declaration of Helsinki, the Guideline for Good Clinical Practice, local laws, code of ethics, etc.

Business rules specified in generic requirements for clinical trials are general and stand for many clinical trials. Business rules stated in clinical trial related rule sources differ in different clinical trials because trials are performed with different purposes, are executed in different countries. However, some rules stated in clinical trial related documents may be used as generic rules with slight variation. For example, the rules for validation of adult patient vitals examination data are similar in various clinical trials. A global view on rules in clinical trials gives an impression that a major part of rules for each trial is general and these rules with nonessential changes may be applied in other clinical trials.

The paper is organised as follows. Section 1 introduces the paper. Section 2 gives a brief overview on clinical trials. Section 3 proposes the way to improve clinical trial design. Section 4 discusses modelling of clinical trials using UML. Section 5 shows the way to retrieve business rules from UML models and presents the prototype of application for business rules management. Section 6 concludes the paper.

2 Clinical trials

When the purpose of the trial is defined, the document used to justify the design and describe the trial procedures in detail is prepared. This document is called a clinical trial protocol. A protocol is the document containing the information relating to the purpose, eligibility criteria, design and conduct of the trial [1], [8]. A protocol also specifies what activities are to be performed in a trial, what measurements are to be evaluated, how the study will be coordinated, etc. [1]. Generally, protocols define all aspects of the proceeding of a particular clinical trial. Thus, it is a crucial document, and if incomplete, disorganised or incorrect, can prejudice the whole study [3].

The analysis of data gathered during the clinical trial is as important as protocol design, because the obtained results are fundamental to subsequent activities. The main purpose of having well designed forms is to make patient evaluations suitable for statistical analysis, but before performing the analysis all data has to be collected, processed and checked [1].

It is obvious that the means for data collection and analysis in clinical trials have to be precise, qualitative, verified and validated. These requirements also stand for the applications used in clinical trials for data interchange, data entry, data clarification, data records tracking, etc. The lack of the system for gathering and managing all the requirements for particular trial complicates the control of quality. Requirements for particular clinical trial may be expressed as business rules. The use of business rules approach principles may facilitate the control of the quality. This is especially applicable to clinical trial applications, as these applications are rules centred. Trial related knowledge and know-how knowledge stored in business rules repository gives a broad view on a whole of all requirements. The question how to gather all the rules in to one repository arises here. We propose to use the model of a clinical trial to gather all the rules in to rules repository. The modelling of a clinical trial may slightly prolong clinical trial design and may require additional resources, but it is definitely advantaged. First of all a graphical model of a clinical trials gives a broad view on the organisation and the procedures of a clinical trial. Besides, clinical trial model is suitable to capture trial related rules. The following sections discuss the use of clinical trial models in detail.

3 Improvement of clinical trial design

Mostly the model of the clinical trial is not created during the design of the clinical trial. As a result of the clinical trial design a clinical trial protocol is produced. The clinical trial protocol presents all the information needed for the conduct of the clinical trial, but the information is represented in natural language. Additional charts and diagrams may be included in clinical trial protocols, but these do not present the conduct of trial as a whole and in detail. The use of natural language for clinical trial description has both negative and positive aspects:

- the positive aspect of the use of natural language is clarity of the protocol for everyone interested in the clinical trial. In other words the protocol is understandable, easy readable and does not require any special knowledge;
- the negative aspect of the use of natural language is ambiguity of natural language. Natural language is informal and can be interpreted. As clinical trial protocol is the primary document for the conduct of clinical trial it is desirable to have unambiguous specification of all trial procedures.

The use of some formal or semi formal modelling language for clinical trial modelling may allow reduce the ambiguity of the protocol. But as there are special requirements for the clinical trial protocol and it has to be approved before the start of the trial, it is impossible to present a model of the trial instead of clinical trial protocol to the responsible authorities. Thus a model of a clinical trial cannot replace protocol. A model of a trial may be prepared in parallel with the construction of protocol instead of replacing the clinical trial protocol with the model of the trial. It would be even better to start the design of the trial from the model, but it may be impossible, because the design of the model may prolong the design of the study. Therefore the trial should be modelled using any formal or semi formal language to represent the procedures of the trial in a graphic way in parallel with the design of the protocol or just after the clinical trial protocol is created.

4 The model of a clinical trial

There are many modelling languages suitable to represent different aspects of systems – UML, IDEF, conceptual graphs, etc [11]. As UML became the most popular modelling language for any kind of systems in recent years, we analyse the use UML for clinical trial modelling in this paper. The Unified Modelling Language is a visual language for specifying, constructing and documenting the artefacts of systems. It is a general-purpose modelling language that can be used with all major object and component methods, and that can be applied to all application domains (e.g., health, finance, telecom, aerospace) [5]. UML diagrams can be classified into three different classes [6]:

- diagrams describing the roles and obligations of system users generally (Use Case diagrams). In the clinical trial models these diagrams should represent the roles and obligations of the clinical trial team members and participants. For example, the right to revoke the patients informed consent or the obligation of investigator to record medical history in the Case Report Form can be represented in the UML Use Case diagrams;
- diagrams describing structural system aspects (class and object diagrams). In the clinical trial model class diagrams should be used to represent the organisation of a trial in detail. For example, each examination, visit, laboratory assessment, etc., should be represented as classes with attributes and operations. Class model may be used to create the structure of the database for the clinical trial data;
- diagrams describing the internal and external behaviour of system (state transition diagrams, sequence and collaboration diagrams). In the clinical trial models these diagrams should be used to represent the sequence of actions in each step of a clinical trial. For example, the proceeding of screening visit can be described in sequence or collaboration diagrams and the states of the patient diary can be represented in state transition diagrams.

UML models are not fully formal [10]. Some information represented in UML diagrams can be interpreted, but generally UML models are suitable for automation of systems development. Business rules representing requirements for clinical trial can be retrieved from UML models and placed in the business rules repository. We highlight the following main advantages of UML usage for clinical trial research:

- UML model would give a broad graphical view on the whole trial. This would improve quality control and documentation of clinical trial procedures;
- Duties and responsibilities of clinical trail team members represented in UML Use Case models would simplify preparation of operational manuals for investigators and other team members;
- The organisation of clinical trial structural components represented in UML Class diagrams, can be for clinical trial database design;

- Representation of all requirements for valid clinical trial data in one model would give a broad view on all rules for data validation.

The results of our previous research showed that UML models could be used to retrieve business rules from UML models. The types of business rules that can be retrieved from UML models we discussed in detail in [9] and they are not analysed in this paper in detail. Further we shortly present the results of he experiment performed to check the reliability of proposed.

5 Retrieval of clinical trial rules

In this section a few sample diagrams are presented to show how different aspects of clinical trials can be represented using UML. We were using Sybase® PowerDesigner® 9.0 for modelling of a clinical trial.

A Use Case model representing the roles and obligations of Investigator and patient at Visit1 is shown in Figure 1. The sample Use Case model is presented only in the scope of Visit 1.

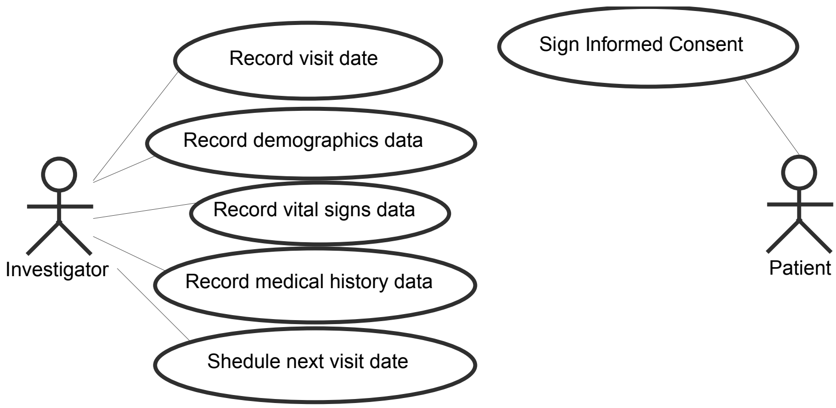


Fig. 1. Use case model representing the roles of Investigator and patient on Visit 1

Figure 2 shows the sequence of actions for Visit 1. Sequence diagram contains rules such as “Data can be collected only after the Informed consent is signed by the patient”.

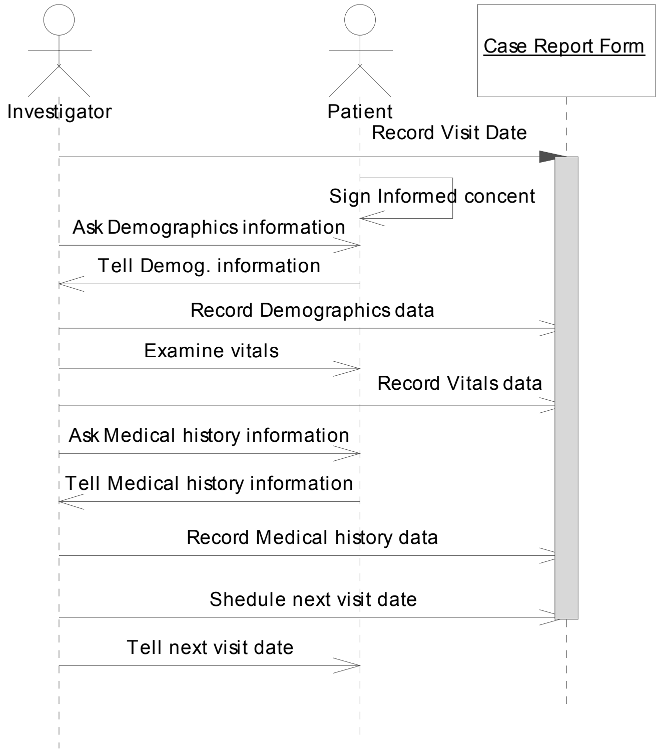


Fig. 2. Sequence diagram representing the sequence of actions to be performed on Visit 1

Figure 3 presents class diagram representing the organisation of objects involved in Visit 1 and their relationships. Class diagram contains rules describing the relationships between classes, class properties constraints. In Figure 3, for example, the rule “At least one medical history record has to be recorded for each patient” is represented as relationship between classes “Medical history” and “Visit 1”.

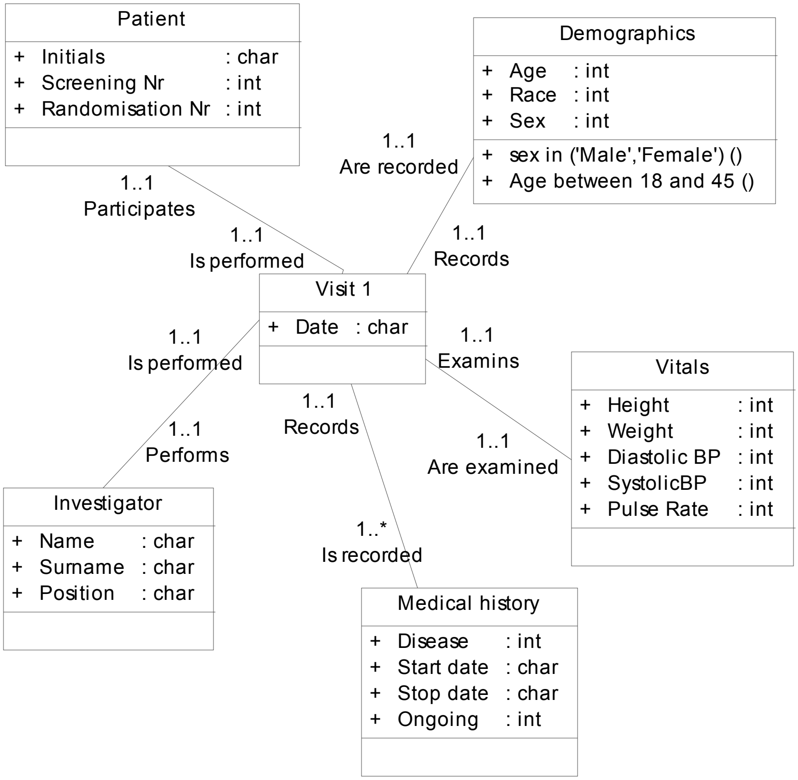


Fig. 3. Class diagram representing the organisation of Visit 1

The prototype application was developed to retrieve business rules from UML models and place in the business rules repository. At this stage of research the prototype application operates only with Use Case diagrams. Retrieval of business rules from other UML diagrams is being implemented.

Created model is stored in XML file. Data are copied from XML file to temporal storage for model analysis. The search for business rules represented in the use case model can be performed directly in XML file, but in order to accelerate the process of business rules search, the data is copied to temporal storage and indexed. Figure 4 presents a general architectural view of the software prototype.

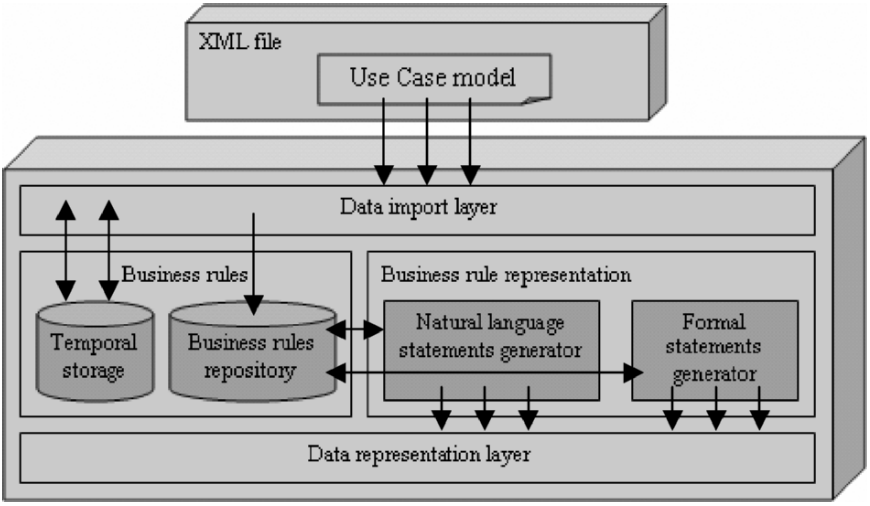


Fig. 4. The architectural view of the experimental tool

The following components of business rules are looked for and copied to temporal storage: actors, use cases, relationships between actors and use cases, stereotypes of relationships. Figure 5 presents a part of the business rules repository structure for storage of rules represented in UML use case diagrams. Table "Actors" is used to store the information of actors represented in use case model. The purpose of table "Use Cases" is to store the information of use cases represented in the use case model of the business system of interest. The information of business systems actor's roles and obligations is stored in the table "Obligations". Table "Predicates" contains the information of predicates, which are formed, on the basis of actor's roles and obligations data. The relationships between business rules components are stored in table "Use_Case_Rule". The information stored in this table is used to express business rules formally and informally. Formal business rules expressions are stored in the table "Formal_UseCase_Rule". Business rules expressed in natural language are stored in table "UseCase_Rule_Text".

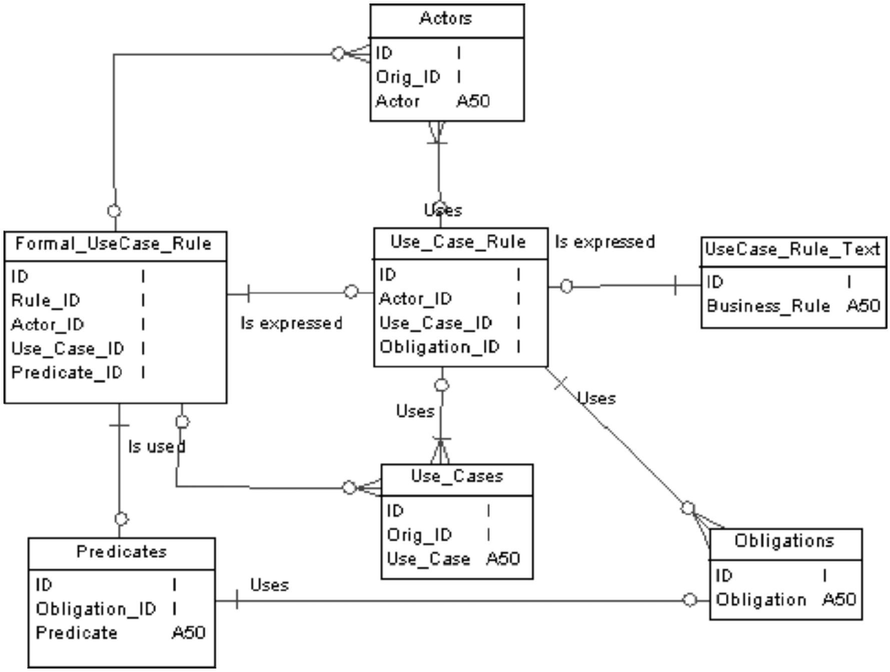


Fig. 5. A part of the business rules repository

Stereotypes of relationships are used to specify the roles of actors. Business rules components are connected and business rules are composed in the following step. The roles of business actors are used to form business rules expressions. Business rules stored in the repository can be expressed in natural language. Figure 6 presents a list of business rules retrieved from Use Case model.

It is natural that the clinical trial protocol does not contain all the requirements for clinical trial conduct, because too much detail clinical trial protocol may exceed the normal ranges of the protocol pages amount. More requirements for particular trial are general and may not be recorded in any document, but these requirements are still active. These requirements come from know-how knowledge. For example, the requirement that each visit date recorded in the CRF has to be correct is obvious, but therefore it has to be checked to avoid human mistakes. Additional trial related requirements described in other documents for management of clinical trial have to be entered into repository manually.

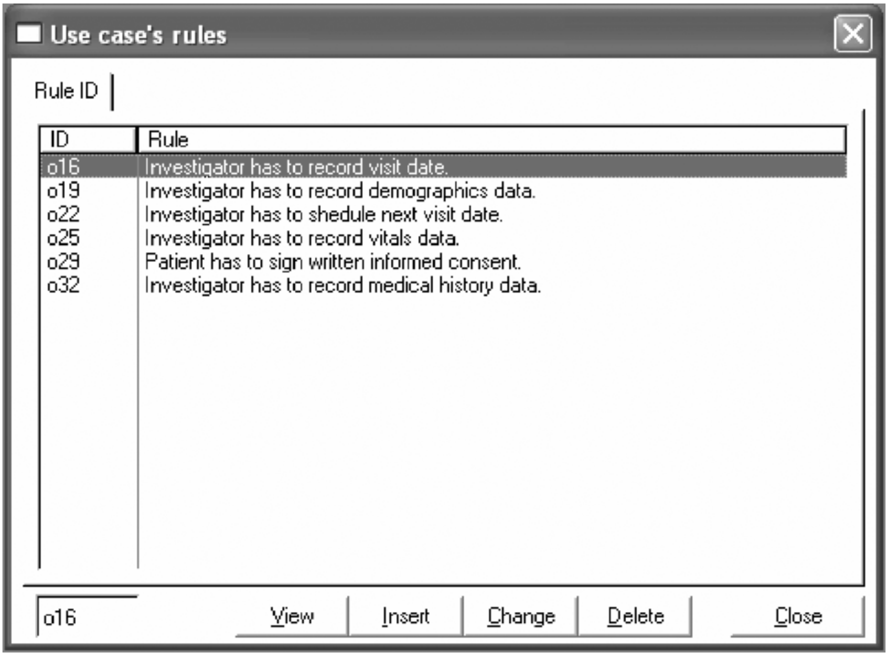


Fig. 6. A list of rules retrieved from Use case model

6 Conclusions

Analysis showed that clinical trials are very important and have to be conducted assuring the quality and precision, because investigations are performed on humans. As there are many requirements for conduct of clinical trials, it is difficult to maintain the quality of the trial. We proposed to use UML for clinical trial modelling in the paper. The benefits of UML model of clinical trial were explored and presented. Analysis and the results of the experiment showed that the design and conduct of a clinical trial could be improved using UML for trial modelling. The future work will concentrate on the implementation of retrieval of business rules from all UML diagrams and development of user interface for additional rules management.

References

1. Ablett S. An introduction to clinical trials. United Kingdom Children's Cancer Center study Group, 2004. URL: http://www.ukccsg UKCCSG Data Centre, [retrieved on 2006.04.05].

2. American Cancer Society. Clinical Trials: What You Need to Know, 2004. URL: http://www.cancer.org/docroot/ETO/content/ETO_6_3_Clinical_Trials_-_Patient_Participation.asp, retrieved on 2006.04.12.
3. Modgil S, Hammond P, Wyatt J C, Potts V. Developing A Knowledge-based Tool for Authoring Clinical Trial Protocols. First European Workshop on computer-based support for clinical guidance and protocols (EWGLP 2000). IOS Press, Amsterdam, Leipzig, Germany, 2000.
4. Nammunil K, Pickering C, Modgil S, Montgomery A and others. Design-a-Trial: A Rule-Based Decision Support System for Clinical Trial Design. Knowledge Based Systems, 17/2-4, 2000, p. 121-129.
5. Object Management Group (OMG). Unified Modeling Language (UML) Specification: Infrastructure version 2.0. URL: http://www.omg.org/docs/formal/05-07-05.pdf, retrieved on 2006.04.14.
6. Shen W, Compton K, Huggins J K. A Toolset for Supporting UML Static and Dynamic Model Checking. In proceedings of the 26th International Computer Software and Applications Conference (COMPSAC 2002), Prolonging Software Life: Development and Redevelopment, Oxford, England, IEEE Computer Society, 2002: p. 147-152.
7. U.S. Department of Health and Human Services, Food and Drug Administration. Guidance for Industry. Computerised systems used in Clinical Trials. Draft guidance, 2004. URL: http://www.fda.gov/Cder/guidance/6032dft.htm, retrieved on 2006.04.17.
8. U.S. Department of Health and Human Services, Food and Drug Administration. Guidance for Industry. E6 Good Clinical Practice: Consolidated Guidance,1996. URL: http://www.va.gov/chrr/regulatory/vrb/files_forms/GoodClinicalPractices.pdf, retrieved on 2006.03.11.
9. Vasilecas O, Lebedys E, Laucius J. Formal methods for representation of business rules specified using UML. // In: R. Simutis (eds.). Proceedings of the International Conference "Information Technologies for business - 2005", Kaunas: Technologija, p. 41-47, 2005.
10. Vasilecas O, Lebedys E. Business rules repository for business rules represented using UML // In R. Rachev, A. Smrikarov (Eds.). Proc. of the International Conference on Computer Systems and Technologies "CompSysTech, 05", Varna, Bulgaria, 16-17 June. 2004, p. II.5-1 – II.5-6.
11. Vasilecas O, Lebedys E. Repository for Business Rules Represented in UML Diagrams // Izvestia of the Belarusian Engineering Academy, 2005, Vol. 1 (19)/2, p. 187-192.

On Some Inferences Based on Stratified Forward Chaining: An Application to E-Government

El-Hassan Bezzazi

IREENAT, University of Lille 2, France bezzazi@univ-lille2.fr

1 Introduction

This paper is twofold. It introduces first an expert system shell whose inference engine is based on stratified forward chaining presented in detail in the paper [1]. The stratified forward chaining (hereafter *sfc*) was proposed as a generalization for the inheritance networks with exception since it allows more than one antecedent in the premises part of a rule. In [2] *sfc* was tuned to what was called specific stratified forward chaining. Precisely, rules which have more literals in their body are said to be more specific than those with a subset of these literals in their body under some conditions. This way, more specific rules can easily be added to the rulebase and will prevail over more general ones in case they have conflicting conclusions. Expert systems built with this tool are rule based and the user interface is web based. The chaining *sfc* allows to manage the application of conflicting rules (i.e. those whose conclusions are opposed) according to the same intuition as the one in inheritance networks and which is expressed by the rule of preemption [5]. The practical use of *sfc* is presented through the three chaining methods: forward chaining, backward chaining and mixed chaining. Our current aim is to make use of these inference mechanisms to help the user in his interaction with public administration to identify in an efficient way the relevant information he needs. For example, when a user applies for some administrative document, the system would help him know the required documents [3]. This is done through a question-response dialog to determine the user profile. This kind of dialog exempts the user from spending time for extracting the relevant information out of the general available documentation. On the other hand, the fact that the user-system interaction is mainly based on yes/no questions makes it particularly well suited for mobile government, when using mobile phones which have, in general, small screens. The GPL programming language PHP [4] was used to implement the inference engines in our prototype. In its second part, this paper considers also a special kind of inference built on *sfc* called deontic inference. The rationale behind this kind of inference is the possible presence of deontic rules which may impose a change in the base of current facts to make it comply with the obligations they prescribe. The paper is organized as follows. In section 2 we present the language used for knowledge representation and in a rather informal way the specific stratified forward chaining. In sections 3 and 4 we define respectively the stratified backward chaining *sbc* and the stratified mixed chaining *smc* which comply with *sfc*. An example of dialog using *smc*. Section 5 begins the second part of this paper by introducing deontic rules and deontic inference as well as a measure to evaluate how far or how close is a situation described by a set of literals to the ideal situation computed by deontic inference. In section 6 we discuss a more general characterization for deontic inference. In section 7 we give some examples of deontic rules and inferences.

2 Knowledge representation, Cautious union and specific stratified forward chaining

A literal is an atom or an atom preceded by ! which represents negation, it is then a negative literal. Currently, an atom is simply a symbol but atoms with more structure such as comparisons could be easily incorporated in the system. The shape of a rule is: $l_1 l_2 ... l_n > l_m$ where $l_1 l_2 ... l_n$ and l_m are literals. Literals are separated by one space or more in the body of the rule and the conclusion is a single literal.

A knowledge base is identified by a name and has a base of rules and a base of facts. The function *lit* identifies and puts in the set *literals* the literals which appear in the knowledge base.

The function *clean* removes from a set of literals all opposing literals. The function *cu* is a non commutative operation that carries out the cautious union of two sets of literals by adding to the first set only those literals in the second set which do not have their opposites in the first set and cleaning the resulting set. $clean(L) = L - \{l, !l : l, !l \in L'\}$, $cu(L) = clean(L \cup \{l \in L' : !l \notin L\})$

The stratification of a rulebase is carried out by the function s. The stratification mechanism consists in computing for each literal of the rulebase its stratum. The stratum of a literal is, roughly speaking, the biggest number of one-step forward chaining that infers it [1]. The stratum of a rule is the biggest stratum of its body. Note that a stratified rulebase is necessarily acyclic when thought of as a graph. We give, in what follows, an informal description for this inference based on two partial orders p_1 and p_2. A rulebase is said to be stratified by the mapping $s : lit(R) \longrightarrow [1, n]$ if and only if for any rule in R, $l_1, ..., l_n > l_m$ we have $s(l_i) < s(l_m)$ and there is no other such mapping $s' : lit(R) \longrightarrow [1, n']$ with $n' < n$. This stratification induces a partial ordering on R defined by $r \, \mathrm{p}_1 \, r'$ iff

$max(s(body(r)) < max(s(body(r'))$. Let $\prec_2$ be the partial ordering that specificity defines, $r \prec_2 r'$ if and only if $body(r') \subseteq body(r)$ and $r \not\prec_1 r'$. We combine these two orders to define a new partial order $\prec_p$ as $r \prec_p r'$ iff $r \prec_1 r'$ and $r' \not\prec_1 r$. In other words, $r \prec_p r'$ iff $r \prec_1 r'$ and either r is more specific than r' or they do not compare to each other. Let $\prec_t$ be any total ordering on R which is consistent with $\prec_p$, *i.e*, if $r \prec_p r'$ then $r \prec_t r'$. Specific *sfc* inference, starting from a rulebase $R = r_1, ..., r_n$ with $r_1 \prec_t ... \prec_t r_n$ and a set of initial facts L, proceeds as follows: Among these totally ordered rules we look for the first rule r which can be fired by L, *i.e.* such that $body(r) \subseteq L$. If such a rule does not exist, the process is done, otherwise the process continues with the new totally ordered rulebase $R - \{r\}$ and the set of facts $L \cup \{head(r)\}$. Notable properties for *sfc* are, given two sets of literals L, L':

- Non-monotony *i.e.*, in general $sfc(L) \not\subseteq sfc(L \cup L')$
- $L \subseteq sfc(L)$
- If L is consistent then so is $sfc(L)$

The function *sfc* computes the literals inferred using stratified forward chaining and retains at the same time the rules which were applied with relevance in the inference, i.e. the rules which were applied and whose conclusion was actually kept (and not rejected during the cautious union by the opposite literal previously inferred). The relevant rules will help us define in the sequel the stratified backward chaining.

3 Stratified Backward Chaining

Backward chaining is a bottom up method used in expert systems. It starts with a literal called a goal to see, in case it is not an initial fact, if there are rules that support it and so on for the literals in the body of these rules which become sub-goals to be confirmed. The goal is confirmed when all the ultimate sub-goals are among the initial facts. Because of the fact that this method is goal-driven, it will not use all the rules that would have been used to check a given goal by forward chaining. However, attention must be paid to possible cycles that may exist in the rulebase to guarantee the termination of the process. Fortunately, this would never be the case as far as we are dealing with stratified rulebases. This is the classical definition of backward chaining with respect to classical forward chaining. The definition we are about to give now of backward chaining which we will call stratified backward chaining (*sbc* in short) is made with respect to *sfc*. Unlike the definition of classical backward chaining which does not use forward chaining, *sbc* does use *sfc*. Indeed *sfc* allows the selection of the relevant rules *i.e.* those which conclusions are inferred with respect to *sfc* and the initial facts. Besides stratified backward chaining uses two functions *sbc* and *prove* which are executed according to a cross recursion (they call mutually each other). In fact, the function *sbc* selects among the relevant rules those which are especially relevant for the goal being processed.

Consider the following fragment of a rulebase which describe documents required to apply for a passport. There is an exception which is stated in case the application is made for a modification, fore example a new born child to be added into the passport. In this case the stamp, denoted by doc_7, is no more required.

```
renewal > passport_application;
passport_modification > passport_application;
passport_a pplication > documents;
documents > doc_1;
.........................................
documents > doc_9;
passport_modification > !doc_7;
```

If we take as initial facts the only fact passport_modification and as a goal the fact doc_7, the classical backward chaining will conclude on a success whereas we would rather like it to conclude on a failure in accordance with sfc. The solution consists in checking before concluding on a success with the classical backward chaining that the rules which made it possible to deduce the goal are relevant for sfc.

```
Function prove(list_of_goals)
if(list_of_goals is empty) then return(true);
else if (sbc(head(list_o f_goals))) return(prove(tail(list_o f_goals)));
else return(false);

function sbc(goal,list_of_relevant_rules)
if (goal in current_facts) then success=true
else if (list_of_relevant_rules is empty) then success=false;
else if (goal!=conclusion(r))
then success=(sbc(goal,tail(list_of_relevant_rules)));
else if (prove(premises(r))) then success=true;
else success=(sbc(goal,tail(list_of_relevant_rules)));
return(success)
```

list_of_relevant_rules is the set of relevant rules in the rulebase for sfc.
current_facts stocks the facts entered by the user and which initially constitute the facts base.

Fact: Given a stratified rulebase, $sbc(goal, list\_of\_relevant\_rules)$ returns a success if and only if $goal \in sfc(current\_facts)$.

4 Stratified Mixed Chaining

Mixed chaining is an inference method using the forward chaining to infer new facts and backward chaining to confirm facts possibly by questioning the user. The stratified mixed chaining carried out by the function *smc* that we present here combines the forward and backward stratified chaining. In general, it works through a dialog with the user during its execution. The dialog relates exclusively to the truth value of certain facts among the initial facts. It is the function *backward_all* which chooses the facts to be questioned on. The function *sfc* is called in the processing of the form submitted as the answer to each question. It is interesting to notice here that the forms sent to the user during this dialog save at the same time the execution environment of the function *smc*. Indeed, the fact that the http protocol is stateless, the html page sent to the user carries as hidden fields all the data necessary to save the recursion context. Questions are put only about positive literals i.e. if it is a negative literal on which the user must be questioned, the question will be put rather on the opposite literal which then gives the answer for the literal in question. The questioning to be sent to the user through an html form is prepared by the function *to_ask_user*.

```
function smc(list_of_rules,goal)
n = 0;
dialog = false;
while(n < number_o f_rulesand!dialog);
r = list_o f_rules[n];if(conclusion(r) = goal)
if((backward_all(premises(r)))) return(true);
n = n + 1;
return(false);

function backward_all(list_of_goals)
dialog = false;
if(list_of_goals is empty) return(true);
elsefact = head(list_of_goals);
if (fact in result) return (backward_all(tail(list_of_goals));
else if (inverse(fact) in result) return(false);
else if (initial(fact) and not (asked(fact) or asked(inverse(fact))))
to_ask_user(fact);
sfc();
dialog = true;
else if(smc(fact)) return (backward_all(tail(list_of_goals)));
else return(false);
```

number_of_rules is the number of rules in the rulebase.
dialog is a boolean variable used to simulate the mutual recursion.
result stocks the facts deduced with the function *sfc*.
initial returns true if the literal occurs only in the premises false otherwise.
asked returns true for a literal which the user was asked about during runtime.

Fact: Consider a session of *smc* which has concluded either on a success or not and let answers denote the set of literals added to current_facts through the session dialog by the user's answers. Given a stratified rulebase, *smc(goal)* returns a success if and only if $goal \in sfc(current\_facts \cup answers)$.

We give here an example of a dialog which the reader will be able to test at the web address: http://droit.univ-lille2.fr/eadministration/exp.php worked out of government texts. The rule renewal out-of-date<2 childs childs<=4 > validity-5-years must be read as follows. If the case is about a renewal of an out-of-date passport since less than two years and that the children must be recorded there and that their number does not exceed four then the validity duration of the passport will be of five years.

emergency > !stamp_60;
emergency > stamp_30;
emergency > validity_6_months;
emergency > documents;
renewal out_of_date<2 childs childs<=4 > documents;
renewal out_of_date<2 childs childs<=4 > docs_childs;
renewal out_of_date<2 childs childs<=4 > valid_5_years;
renewal out_of_date<2 childs !childs<=4 > documents;
renewal out_of_date<2 childs !childs<=4 > docs_childs;
renewal out_of_date<2 childs !childs<=4 > demand_elders;
renewal out_of_date<2 childs !childs<=4 > valid_5_years;
renewal out_of_date<2 !childs > documents;
renewal out_of_date<2 !childs > validity_10_years;
renewal !out_of_date<2 > first_demand;
modification > documents;
modification > !stamp_60;
modification > current_validity;
documents > form;
documents > 2_photos;
documents > stamp_60;
documents > proof_residence;
documents > identity;

The following rules are used to define the final facts among which the inferred facts should be chosen for the answer.

form > goal; 2_photos > goal;
stamp_60 > goal; stamp_30 > goal;
proof_residence > goal;
identity > goal;
first_demand > goal; demand_elder > goal;
documents_childs > goal;
valid_10_years > goal; valid_5_years > goal;
valid_6_months > goal;
current_validity > goal;

Example of session:
emergency ?
no
renewal ?
no
out_of_date<2 ?
yes
childs ?
yes
childs<4 ?
yes

5 Deontic Rules

Deontic logic is a logic for reasoning about ideal and actual behaviour [11]. Applications of deontic logic can go beyond legal analysis and legal automation to cover other domains like the specification of security policies and fault tolerant systems [10]. Traditionally, this logic is developed as a modal logic with, essentially, a modal operator o to define obligation which can be used in its turn to define permission and prohibition. However Instead of using such an operator and in order to stay at a propositional level, we enrich the language by considering a new kind of literals which are intended, if inferred, to describe an obligation to be respected. These literals, we shall call deontic literals, are of the form $o(l)$ or $!o(l)$ where l is a descriptive literal, *i.e.* a literal defined as in section 2. Let O be the set of deontic literals and Lit_D the set of obligated literals *i.e.* $l \in Lit_D$ if and only if $o(l) \in O$. The function ω returns the obligated literal of a deontic literal *i.e.* $\omega(o(l)) = l$ and the function o returns the deontic literals of a set of literals, $o(L) = L \cap O$. An operation of cautious union *cu'* is defined for two sets of deontic literals L_D, L'_D in the following way. $cu'(L_D, L'_D)$ adds to the first set L_D only those literals in L_D which do not have their opposites in the first set or which do not oblige a literal whose opposite is obliged in L_D.

A deontic rule is a rule where the body is made out of descriptive literals and the conclusion is a deontic literal. Therefore, we shall deal from now on with two kinds of rulebases: A descriptive rulebase R and a deontic rulebase D. Note that the set of rules $R \cup D$ is stratified since R is stratified and the heads of the newly added rules never occur in the rules body.

5.1 Deontic inference and Legal sets

Given a set of literals L, we define elementary deontic inference through *sfc* $edi(L, R \cup D) = sfc(L, R \cup D)$. In the sequel, we shall drop the parameter $R \cup D$ from both $edi(L, R \cup D)$ and $sfc(L, R \cup D)$ in order not to encumber the notation. The result of this inference is a set of literals L' which may contain deontic literals. A set of descriptive literals L is said to be legal or equivalently D-consistent when $sfc(L)$ does not contain a positive deontic literal which is violated by one of its descriptive literals or conflicted by a contrary obligation: L is legal if and only if $o(l) \in sfc(L) \Rightarrow !l \notin sfc(L) \wedge o(!l) \notin sfc(L)$. A set of literals which is not legal is called illegal. inference from an illegal set L yields an illegal set L' since $L \subset L'$. An elementary legalization $\lambda(L)$ for L is an update of L by replacing l with $!l$ whenever $l \in L$ and $o(!l) \in sfc(L)$, in other words $\lambda(L) = cu(\omega(o(sfc(L))), L)$.

The legalization of an illegal set L consists in a succession of elementary legalizations. In order to be able to cumulate along this process the deontic literals in accordance with their cautious union, we need to keep track of them.

$$\begin{cases} \lambda^{(0)}(L) = L, \Delta^0 = \varnothing \\ \lambda^{(n+1)}(L) = cu(\omega(\Delta^{(n+1)}), \lambda^{(n)}(L)), \Delta^{(n+1)} = cu'(\Delta^{(n)}, o(sfc(L))) \end{cases}$$

Since the set Lit of all literals is finite, the sequence $\{\lambda^{(n)}(L)\}_n \in N$ is such that there is necessarily an integer i satisfying $\lambda^{(i)}(L) = \lambda^{(i')}(L)$ for some $i' > i$. Let k and k' be the smallest integers satisfying $\lambda^{(k)}(L) = \lambda^{(k')}(L)$ with $k' > k$. Two cases are to be distinguished:

1) The case where $k = k'$ which means that the sequence $\{\lambda^{(n)}(L)\}_{n \in N}$ is stationary. We define then the deontic inference of L to be $di(L) = edi(\lambda^{(k)}(L))$

2) The case where $k \neq k'$ which means that the sequence $\{\lambda^{(n)}(L)\}_{n \in N}$ is cyclic.

This will be considered as an inconsistency and the deontic inference is undefined $di(L) = \perp$. This means that the set of deontic rules D does not provide the means for handling the case described by L.

Deontic inference defined this way incorporate defeasible reasoning thanks to stratified forward chaining which operates at the descriptive level as well as at the deontic level. The issue of dealing with conflicting and conditional obligations in deontic logic at the light of non-monotonic reasoning has been discussed in several papers [6][8][9]. In particular, in [8] the author recommends the use of an already existing non-monotonic logic with a deontic logic instead of a built-in non-monotonic deontic logic and in [9], the author underlines the difference between an obligation which is defeated and an obligation which is violated. The elementary legalization plays a crucial role in deontic inference. it takes the deontic output of sfc and reuses it to correct the input, possibly altering it by replacing some of its literals by their opposites, to make it comply to the obligations of the rulebase. Making a difference between the declarative statements and the norms is an approach already singled out in [7] where the set of conditional norms is seen as a black box which transforms the input into output following some basic rules.

5.2 Legality Degrees

Let us consider now the case where $di(L) \neq \perp$. $di(L)$ describes the ideal situation to which L must comply whereas $sfc(L)$ describes the actual situation. The similarity between the actual situation and the ideal situation can be evaluated with well known similarity measures for finite sets such as the Jaccard measure. We define the degree of legality of a set of literal L with respect to the rulebases R and D as

$$\delta(L) = \frac{|\, sfc(L) \cap di(L)\,|}{|\, sfc(L) \cup di(L)\,|}$$

.

5.3 A more general characterization for legal sets

Let L, R, D, Lit be respectively a set of literals, a descriptive rulebase, a deontic rulebase and the set of all literals appearing in these sets. Let Lit_D be the set of obligated literals *i.e.* $l \in Lit_D$ if and only if $o(l) \in Lit$. The issue we investigate in this section is, given a consistent set of literals L, in case it is not legal, which legal sets of literals, if any, could be proposed as “legalizations” for it. A model for L with respect to $R \cup D$ is defined as being any consistent set $L = cu(L_D, L)$ of literals with $L_D \subseteq Lit_D$ such that:

(1) $sfc(L)$ is D-consistent and

(2) for any $l \in L_D$ there is $L_l \subseteq L_D$ such that $l \in \omega(sfc(L_l))$ where $L_l = cu(L_l, L)$

Let L be a consistent set of descriptive literals. The legalized set obtained by deontic inference when defined is a model for L.

Proof. The legalized set is $L = \lambda^{(k+1)}(L) = cu(\omega(\Delta^{(k+1)}), \lambda^{(k)}(L))$ for some $k \in N$. We put $L_D = \Delta^{(k+1)} = cu'(\Delta^{(k)}, o(sfc(L)))$. L is D-consistent since it is the legalization of L by di. On the other hand, property (2) is satisfied since the sequence $(\Delta)_{i\in[0,k]}$ is an increasing chain. Ω

Consider the following rulebases:
a>o(c);
a>o(!c);
There is no model for $\{a\}$.
a>o(!b);
b>o(!a) ;
$\{a,!b\},\{!a,b\}\, and\, \{!a,!b\}$ are models for $\{a,b\}$. Indeed, the last one for example is the result for $cu(\{!a,!b\},\{a,b\})$, on the other hand $sfc(\{!a,!b\})=\{!a,!b\}$ and $!a\in sfc(\{!a,b\})=\{!a,b,o(!a)\}$ and $!b\in sfc(\{a,!b\})=\{a,!b,o(!b)\}$

5.4 Examples

We comment in this paragraph some examples on the use of deontic inference and legal sets.

No smoking

Consider a building where smoking is forbidden. However, smoking is allowed in the room number 4 of this building.
room4 > building;
building > o(!smoking);
room4 > !(o(!smoking));
smoking > pollution;
The sets {room4, smoking}, {room4, !smoking}, {building, !smoking} and {room4, building, smoking} are legal sets. The set{building, smoking} is an illegal set and its legalization yields {building, !smoking}.

Consider the set $L=\{building, smoking\}$. $sfc(L)$ is not D-consistent. Consider the set $L=cu(!smoking, L)=\{building, !smoking\}$.

$sfc(L)=\{building, !smoking, o(!smoking)\}$ is D-consistent.

Privacy

Let bob_privacy denote personal data of Bob. Agency A is not allowed to have access to Bob privacy unless Bob is okay. If it happens that Agency A did have access to Bob privacy, it is committed not to communicate it to others. Note that as long as the information cannot be erased from his mind, there must be a rule in the rulebase asserting that no obligation can be made to change this state. Corrective actions as sanctions may be prescribed in this case. This is an example of the so-called contrary-to-duty norms in deontic logic where violation of some obligations must be tolerated in favour of other appropriate obligations which become operative.

Agency_A > o(!bob_privacy);
agency_A bob_okay > !o(!bob_privacy);
agency_A bob_privacy > !o(!bob_privacy);
agency_A bob_privacy > o(!communicate);
agency_A bob_privacy !bob_okay > o(sanction);

Cyclic

A simple example of a cyclic case is the following one where D states that light must be on if it was off and the switch pressed and similarly must be off if it was on and the switch pressed.

!light press > o(light);
light press > o(!light);

Considering $L = \{light, press\}$ as our starting set of literals, the legalisation of L by the first elementary deontic inference gives $\{!light, press\}$, the following step gives $\{light, press\}$ which shows the starting of a cyclic inference. This is explained by the fact that the literal press hides some crucial information to put the light on or off. Actually it should be replaced by two literals $press_0$ and $press_1$ for stating respectively the facts of putting off and putting on the light. There is no model for L. This example shows a need to enrich the language by the concept of action to deal with discrete event systems.

6 Conclusion

The first purpose of this paper was to introduce an effective implementation of inference engines based on a non-monotonic logic. Our system enjoys of the conceptual simplicity of systems based on propositional logics. As a matter of fact the examples given in the literature to introduce and motivate the concerns of non-monotonic logics are mainly written in a propositional language, this is why we restrict our definitions to propositional logics in addition to the fact that interesting classical expert systems based on propositional logics do exist and are effectively used. However, we project to investigate on the use of a kind of first order predicate logics as in Prolog clauses for writing the knowledge base rules. A methodology should also be investigated to help define the specificity of rules with respect to each other during the rulebase construction. The second part of this paper investigated the use of deontic literals to handle normative statements. It was noticed that a set of literals may have several sets as candidates for its legalization in the interpretation structure and that the legalization preferred in some cases by deontic inference was severe in comparison to other possible legalizations. This issue must be investigated further in order to make the interpretation more adequate for a possible completeness theorem to hold. Another promising issue is to consider the concept of action in the language as long as the system encompasses the concept of change in the inferred literals.

References

1. Bezzazi, E-H, Revision and Update based on Stratified Forward Chaining in Frontiers in Belief Revision, edited by Mary-Anne Williams and Hans Rott, 2001 Kluwer Academic Publishers. (2001) 315–332
2. Bezzazi, E-H, Specific Stratified Forward Chaining, Proceedings of the International Conference on Artificial Intelligence, Las Vegas, Nevada, USA, (2000) 1455–1460
3. Bezzazi, E-H, Workflows et Systèmes experts dans l'administration électronique, Communication au Colloque International Administration électonique et qualité des préstations administratives, Lille, France, 11/19/2004
4. The official site of PHP: http://www.php.net
5. Horty, J., Thomason, R., Touretzky, D., A skeptical theory of inheri-tancein nonmonotonic networks, Artificial Intelligence, (1990) 311–348
6. Horty, J., Nonmonotonic foundations for deontic logic, In Nute, D., ed., Defeasible Deontic Logic. Kluwer. 17–44
7. Makinson D., Van Der Torre L., Input-output logics, Journal of Philosophical Logic, vol. 29, 2000, p. 383–408
8. Prakken, H., Two Approaches to the Formalisation of Defeasible Deontic Reasoning, Studia Logica 57 (1), pp. 73–90, 1996
9. Van Der Torre L., Violated obligations in a defeasible deontic logic, In A.Cohn (ed.), Proceedings of the Eleventh European Conference on Artificial Intelligence (ECAI'94), pages 371–375, John Wiley & Sons, 1994
10. Wieringa R.J., MeyerJ.-J.Ch., Applications of Deontic Logic in Computer Science: A concise Overview, In Deontic Logic in Computer Science, pp 17–40, John Wiley & Sons, Chichester, England, 1993
11. Von Wright G.H., An Essay in Deontic Logic and the General Theory of Action, Acta Philosophica Fennica, Fasc. 21. North-Holland, 1968

Requirements Determination for Knowledge Management Systems in Information Technology Outsourcing Relationships

Dolphy M. Abraham

College of Business Administration, Loyola Marymount University, 1 LMU Drive, MS-8385, Los Angeles, CA 90045, USA

Abstract: This paper discusses ongoing research studying requirements determination for knowledge management (KM) systems across organizational boundaries. The domain of information technology outsourcing (ITO) relationships is used as the example in this discussion. Business networks, such as, outsourcing arrangements for information technology management or business process management have become a routine arrangement. Data and knowledge sharing becomes a vital aspect of operating and managing such business networks. Consequently, the role of knowledge management becomes more important as business networks become more prevalent. Knowledge management, in the context of business partnerships or networks, is the set of processes necessary to facilitate sharing of information and process knowledge. KM processes also direct the information and knowledge sharing in a way that supports each organization's objectives.

The relationship between the client and service provider in an ITO engagement is unique with respect to the types of processes, information technology infrastructure, geographical location, terms of the contract and individual managers involved. The study of ITO engagements also points out that information and knowledge sharing is critical during the various stages of the engagement. At the start of the ITO engagement, the service provider is dependent on the client for process knowledge relating to the tasks under the contract. During the contract period and at the end, the client is dependant on the service provider to learn about changes in the business environment observed by the service provider's staff. Each of these dependencies involves the use of the appropriate systems or procedures to share the relevant information and process knowledge. These systems and procedures also have to contend with local social and business cultures in the various locations of the client and service provider organizations. This paper identifies the key issues that affect requirements determination for KM systems that support the various stages of ITO relationships.

1 Introduction

Knowledge Management is the organizational process for acquiring, organizing and communicating the knowledge of individual employees so that the work of the organization becomes more effective (Alavi and Leidner 1999). Knowledge Management (KM) is an increasingly important process in business organizations because “managing human intellect – and converting it into useful products and services – is fast becoming the critical executive skill of the age” (Quinn et al. 1998). Grover and Davenport (2001) state that KM becomes “an integral business function” when organizations “realize that competitiveness hinges on effective management of intellectual resources.” Grover and Davenport (2001) also argue that knowledge management works best when it is carried out by all the employees of the organization and not just KM specialists.

Business organizations frequently partner with other firms to complement their core competencies. To collaborate effectively, partner firms have to communicate with each other information about business processes as well as share ideas of how to design or improve business processes. This phenomenon of knowledge sharing across organizational boundaries is called interorganizational learning (Argote 1999). Knowledge management is the set of processes necessary to facilitate interorganizational learning and direct it in a way that supports the organization’s overall objectives.

Business networks, such as, IT outsourcing and business process outsourcing to multiple service providers have become a routine option in both product-oriented and service-oriented industries. Data and knowledge sharing becomes a vital aspect of operating and managing such business networks. Consequently, the role of knowledge management becomes more important as business networks become more prevalent.

We propose a framework that will help plan the requirements for systems that support knowledge management activity in inter-organizational relationships. The framework addresses a key need for organizations as they seek to manage their operations in a manner different from traditional methods. As organizations rely increasingly on partners to carry out some of the business processes, the ability to share not only operational data but also process knowledge between partners becomes a critical capability. By helping determine the requirements for knowledge management systems, the proposed framework will support the successful implementation of outsourcing relationships.

2 Prior Research

Prior research in KM originates in the area of organizational learning. Subsequently the area of knowledge management developed as information technology (IT) tools to support document and content sharing became widely available. We first discuss prior research into KM processes within organizations followed by the experience with KM initiatives across organizational boundaries as well as management issues that arise in IT outsourcing relationships.

2.1 Knowledge Management Processes within Organizations

Software applications called Knowledge Management Systems (KMS) are tools used to support the Knowledge Management process by providing document repositories, employee directories showing areas of expertise, discussion groups and information retrieval tools (Davenport et al. 1998). Where such applications have been deployed, it has been noted that many such initiatives have failed due to several factors including neglect of people management issues, and because KM initiatives are seen as information technology (IT) projects rather than initiatives that deal with changes in organizational processes (Storey and Barnett 2000). In order to implement KM practices in infrastructure management, it is necessary to understand the processes involved in knowledge management. Abraham (2000) posits that the steps in capturing and using knowledge in an organization consist of the following processes linked as shown in Figure 1.

- Collection or acquisition of content
- Organization or filtering of the document repository
- Analysis of the repository
- Utilization of the contents for subsequent work processes

Collection or *acquisition* of content involves the accumulation of content from all relevant business processes in any form that can be documented. This can include formal (or finished) documents, working papers or notes, e-mail messages, archives of instant messages, etc. The accumulation of various types of content into a comprehensive archive that can be searched by all authorized users facilitates the search for prior work product. Each type of document can be categorized by project, incident, or the employees involved. Meta-information in the form of categories and keywords facilitate the task of retrieval. Such archives can also include links to external sites where appropriate.

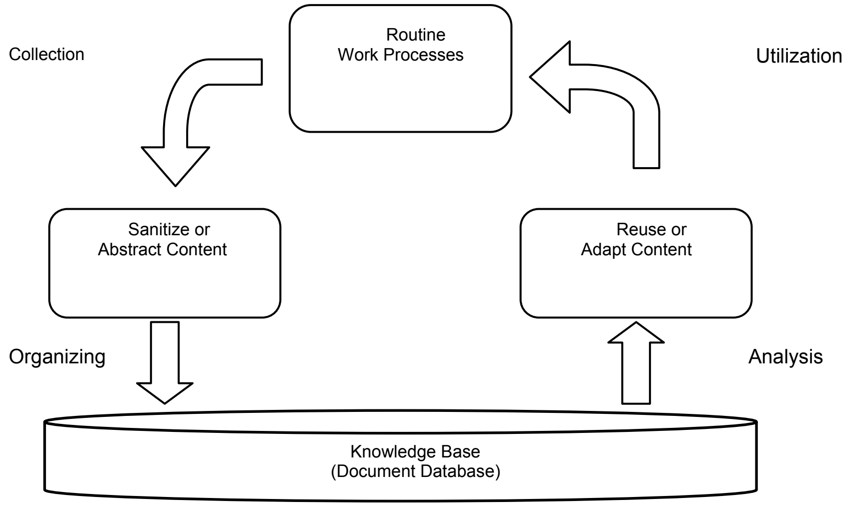


Fig. 1. Linking the KM Processes within an Organization

Organizing or *filtering* helps to make the archived content more usable. As the volume of content in the repository increases, it becomes necessary to make the task of searching through the collection more effective. Database capabilities become necessary to manage the documents. The meta-information that was developed while the documents were accumulated help to structure the document database. Information about domain specific and organization specific processes can also help to structure the document database. Other ways in which documents can be categorized include nature of usability (reusable, adaptable, training material, etc.) and usage patterns (stage of project, users, organizational unit, etc.).

Analysis of the repository helps determine relationships between content items. For instance, it is possible that multiple projects use similar skills, information or processes. As organizations become larger it becomes more difficult for project teams to coordinate with each other. By analyzing document content from across the organization and creating a lexicon of work-related terms or concepts, it becomes possible to add value to the document database. The analysis will help identify relationships between documents and also identify different types of relationships (common skills, personnel, or resources). Items throughout the repository can be analyzed and it is conceivable that links to sources outside the document database can be added based on terms in the lexicon.

Utilization is how the organization derives benefit from the KM process. Making use of the content from the document database is the initial step. Making the use of documents from the repository a standard step in any project needs to become the practice in the organization. By incorporating the use of the accumulated knowledge a part of the routine work process ensures that all employees understand the value of the KMS and how it can be used on a daily basis.

In order to implement this set of processes it is necessary to understand the organizational culture and how it will impact the KM initiative. It is necessary to plan properly and implement the system in layers so that capabilities become available when users are ready to make use of them (Abraham and Seal 2001). Financial and performance metrics are needed to evaluate the KM initiative. (Abraham et al. 2004).

Managing both documented and undocumented (tacit) knowledge is critical (Polanyi, 1967). Managing the documented knowledge involves determining which formal documents must be shared and the mechanisms that must be set up to make them accessible. Identifying any existing mechanisms for information and idea sharing within an organization as well as between organizations will help determine the processes used to share tacit knowledge. The current practices must be extended to enable decision makers within the organization as well as those at interdependent infrastructures to learn from prior incidents. The framework must also enable an individual decision maker to get in contact with another individual who may have situation-specific knowledge. Nidumolu et al. (2001) report that "in order to be successful," (KM initiatives) "need to be sensitive to features of the context of generation, location and application of knowledge." The operating procedures, technology and information systems and organizational culture will collectively impact KM efforts in infrastructure management.

2.2 Knowledge Management across Organizational Boundaries

Prior work in the area of interorganizational learning (IOL) includes studying the role of information technology to support interorganizational learning. Scott (2000) studied the process of and reasons for information technology (IT) support for interorganizational learning. Studying the disk drive industry, Scott identified the need for interorganizational learning to help "cope with the complexity of new products and the capital intensity" in the industry. The model developed as a result of the study helps explain the role of IT in lower and higher levels of interorganizational learning. Gottschalk (2001) surveyed law firms in Norway to determine the predictors of information technology support for interorganizational learning. Firm cooperation and knowledge cooperation were identified as significant predictors. Inter-organizational trust was another factor that affected the use of IT to support interorganizational knowledge management.

Other studies have focused on organizational factors that affect knowledge management across organizational boundaries. Abou-Zeid (2002) studied the factors that impact knowledge transfer between subsidiaries in multi-national corporations. A condition necessary for successful knowledge transfer between subsidiaries was identified to be the explication of the value systems of the subsidiaries involved in the knowledge transfer. A list of organizational values was used to develop an ideal organizational value profile for inter-organizational knowledge transfer. Learning-by-hiring was another strategy for acquiring knowledge studied by Song et al. (2003). Studying the patenting activities of engineers who moved from U.S. to non-U.S. firms, they suggest that "learning-by-hiring can be useful when hired engineers are used for exploring technologically distant knowledge rather than reinforcing existing firm expertise, and also for extending the hiring firm's geographic reach. Hansen (2002) introduces the concept of knowledge networks. His study showed that project teams obtained more knowledge and completed projects faster when the knowledge transfer utilized shorter paths between organizational units. He identified the need for additional research studying the impact of connections between organizational units and the relationship between knowledge content in different organizational units.

2.3 Information Technology Outsourcing

Information Technology Outsourcing (ITO) is broadly defined as a process undertaken by an organization to contract-out or to sell the organization's IT assets, staff and/or activities to a third party supplier who in exchange provides and manages IT assets and services for monetary return over an agreed period of time (Kern et al. 2002). In this contractual relationship the outside vendor assumes responsibility of one or more IT functions, but alternative functions may require alternative processes.

Variations on ITO engagements can include technical staff placed within the client organization to application development with staff located locally or at remote locations, and even dedicated offshore development centers. Offshore Development Centers (ODC) are comprised of a facility and staff dedicated for a client's IT project requirements and are a virtual extension of client's development environment. The ODC team of developers and project managers links up to client's offices and machines via dedicated links and thereby works in the same environment as the client's IT team.

To make the decision about outsourcing it is necessary to understand the factors that affect ITO engagements. These include globalization and the factors that impact globalization. The presence of IT infrastructure especially high-bandwidth network infrastructure is a necessary component for offshore engagements. Regulations and community activism are also factors likely to influence companies faced with the decision to outsource IT processes.

Once the decision to outsource has been made, the success of the engagement is dependent on multiple risk factors (Kirkwood, 2005), including:

- Conflict resolution mechanisms
- Control over the engagement including control over intellectual property and human resources
- Communication mechanisms between managers, between locations, and across management layers
- Management and measurement methodologies including service level agreements, governance structures, and meeting schedules.

Process and governance design becomes the means by which risks can be managed during the life of the engagement. Many managers in client organizations believe that the outsourcing vendor should have processes in place for service delivery and support, removing the need for internal process development. It is vitally important, however, for process and governance design to be driven by the client organization itself in order to ensure alignment with the IT services that support core processes (Kirkwood, 2005). Effective governance procedures require timely sharing of information related to the performance of the service provider.

The study of ITO engagements also points out that information and knowledge sharing is critical during various other stages of the engagement. At the start of the ITO engagement, the service provider is dependent on the client for process knowledge relating to the tasks under the contract. During the contract period and at the end, the client is dependant on the service provider to learn about changes in the business environment observed by the service provider's staff. Each of these dependencies involves the use of the appropriate systems or procedures to share the relevant information and process knowledge. These systems and procedures also have to contend with local social and business cultures in the various locations of the client and service provider organizations (Abraham et al. 2005).

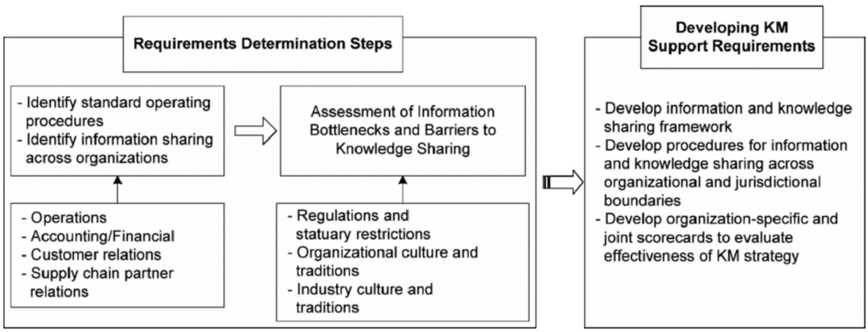


Fig. 2. Requirements Determination for Interorganizational KM Systems

3 Requirements Determination for Inter-organizational KM Systems

Unlike traditional systems analysis processes, determining requirements for KM systems being used in an inter-organizational setting necessitates multiple levels of analysis. There is a need for business process analysis followed by the more detailed analysis of systems requirements. In the context of IT outsourcing relationships, this implies the need to understand the higher level processes first before studying the detailed information and process knowledge that is shared between the service provider and the client.

The first task is a business process analysis of organizational partnerships focusing on the information and knowledge sharing mechanisms already in place. In addition it is necessary to understand the standard operating procedures in place at both the service provider and client organizations. These processes are likely to be those that are prescribed in the outsourcing contract and are part of the operating arrangement. In a sense, this analysis identifies the procedures that generate information and process knowledge that need to be communicated and the planned communication mechanisms between the two organizations.

This task starts with an assessment of operating procedures and the corresponding information systems and procedures associated with tasks that span organizational boundaries as well as tasks within one organization that have a significant impact on the other organization(s) in the partnership. Such systems and procedures would be situated in the day-to-day operations and management of each company, including accounting and financial, customer relations, and outsourcing contract arrangements. The information systems involved with the delivery of the primary product or service would also be part of this analysis. The purpose is to determine how these existing systems are used in a knowledge management role. The questions that will be addressed include how information is shared across the organization in each company and the interaction between decision makers across the organization. In addition, the analysis will help determine the existing formal and informal procedures used within each organization to learn from prior experience.

The second task as part of the analysis involves identifying the factors that affect sharing of information and process knowledge. This task involves determining how the organization culture and traditions affect the practice of knowledge sharing. Regulations and jurisdictional variations of regulations as well as industry practices that impact knowledge sharing will be studied. These factors affect information and knowledge sharing across the organizations in a partnership or network. Regulations and industry practices also impact how managers in each organization use information from other organizations. This analysis will help identify interdependencies between the companies in decision-making processes.

A critical factor in this step of the analysis is in understanding the impact of cultural differences especially those related to the countries in which the two companies have their primary locations. Understanding the cultural factor helps in adapting KM processes to each partner organization's expectations of how information sharing needs to be carried out.

Based on the analysis of the current state across multiple organizations and partnerships, the knowledge management support strategy can be developed. The primary component of the strategy is a general framework that can be used to plan and execute KM initiatives. This includes the processes and systems support necessary to facilitate the sharing of information and process knowledge between the service provider and the outsourcing client. These processes have to be specified to account for the interaction across corporate boundaries and needs to take into consideration the location, language differences and cultural differences between the two organizations.

To manage the KM initiatives, it is also necessary to identify metrics that can be used to measure the use and effectiveness of the KM processes across the organizations in the partnership. These metrics will help evaluate processes within a single organization as well as process that span multiple organizations.

4 Next Steps

The proposed framework will help develop systems to support knowledge management across organizational boundaries. The next step in this project will involve using this framework in real-life outsourcing relationships to test the applicability of the framework to actual settings. This will be attempted in the context of multiple outsourcing projects involving service providers and client organizations located in different countries. In parallel, outsourcing partnerships between companies in the same country will be used to apply the framework. The application of the framework in cross-border settings will be compared with IT outsourcing relationships within one country to identify factors that are specific to cross border relationships.

The expectation is that the final framework following refinements based on real-life testing will enable organizations to create effective KM systems that enhance the quality of outsourcing relationships.

References

Abou-Zeid, E., (2002) An Ontology-based Approach to Inter-organizational Knowledge Transfer, *Journal of Global Information Technology Management*, Vol. 5, No. 3, pp. 32-47.

Abraham, D.M. (2000) An Information Systems Framework for the Development and Deployment of a Knowledge Management Infrastructure, *Proceedings of the INFORMS/KORMS International Conference*, pp. 1280-1284, Seoul, Korea, June 2000.

Abraham, D.M., Brownstein, E., Dai, W., Golper, S., Luolamo, T., Merter, Y., Moalej, A., and Ruvalcaba, C. (2005), Planning Successful I.T. Outsourcing Engagements: Lessons Learned from East Asian Service Providers, *Proceedings of the First International Conference on Managing Globally Distributed Work*, pp. 267-272, Bangalore, India, December 2005.

Abraham, Dolphy M., Fuentes, C. and Abraham, Dulcy M., Evaluating Web-Based Bidding in Construction: Using Simulation as an Evaluation Tool, *International Journal of Electronic Business*. Vol. 2, No. 2, pp. 121-141, March-April 2004.

Abraham, D.M. and Seal, K.C. (2001) CLASIC: Collaborative Layered System using Intranet Capabilities, *Logistics Information Management*, Vol. 14, No. 1&2, pp. 99-106.

Alavi, M. and Leidner, D.E. (1999), Knowledge Management Systems: Issues, Challenges and Benefits, *Communications of the AIS*, Vol.1.

Argote, L. (1999) Organizational Learning: Creating, Retaining and Transferring Knowledge, Kluwer, Norwell, MA.

Davenport, T.H. De Long, D.W. and Beers, M.C. (1998) Successful Knowledge Management Projects, *Sloan Management Review*, Winter, pp. 43-57.

Gottschalk, P. (2001) Predictors of Information Technology Support for Interorganizational Knowledge Management: Lessons learned from Law Firms in Norway, *Knowledge and Process Management*, Vol. 8, No. 3, pp. 186-194.

Grover, V. and Davenport, T.H. (2001) General Perspectives on Knowledge Management: Fostering a Research Agenda, *Journal of Management Information Systems*, Vol. 18, No. 1, Summer 2001, pp. 5-21.

Hansen, M.T. (2002) Knowledge Networks: Explaining Effective Knowledge Sharing in Multiunit Companies, *Organization Science*, Vol. 13, No. 3, pp. 232-248.

Kern, T., Willcocks, L. P., Heck, E. V. (2002), The Winner's Curse in IT Outsourcing: Strategies for Avoiding Relational Trauma, California Management Review, Vol. 44, No. 2, pp. 47-69.

Kirkwood, S. (2005), Intelligent Outsourcing: Applying ITIL Process Governance and Architecture to Outsourced IT Operations, International Network Services.

Nidumolu, S.R., Subramani, M. and Aldrich, Alan, (2001) Situated Learning and the Situated Knowledge Web: Exploring the ground beneath Knowledge Management, Journal of Management Information Systems, Summer, Vol. 18, No. 1, pp. 115-150.

Polanyi, M. (1967) The Tacit Dimension, Doubleday, Garden City, NY.

Quinn, J.B., Anderson, P. and Finkelstein, S. (1998), Managing Professional Intellect: Making the Most of the Best, *Harvard Business Review of Knowledge Management*, HBS Press, pp. 181-205.

Scott, J. E., (2000) Facilitating Interorganizational Learning with Information Technology, *Journal of Management Information Systems*, Fall 2000, Vol. 17, No. 2, pp. 81-113.

Song, J., Almeida, P. and Wu, G. (2003) Learning-by-Hiring: When is Mobility More Likely to Facilitate Interfirm Knowledge Transfer? *Management Science*, Vol. 49, No. 4, pp. 351-365.

Storey, J. and Barnett, E., (2000) Knowledge Management Initiatives: Learning from Failure, *Journal of Knowledge Management*, Vol. 4. No. 3.

Using Concept Maps in Adaptive Knowledge Assessment

Alla Anohina, Vita Graudina, Janis Grundspenkis

Riga Technical University, Kalku street 1, Riga, Latvia, LV 1658
{alla.anohina; vita.graudina; janis.grundspenkis}@cs.rtu.lv

Abstract: The paper presents a novel approach regarding adaptive knowledge assessment using concept maps. Adaptive knowledge assessment adapts assessment tasks to the ability and knowledge level of a particular learner and makes a corresponding system more powerful and valuable. The already developed prototype of an intelligent knowledge assessment system based on multiagent paradigm and concept maps is described, but mainly the paper focuses on the support of adaptive assessment regarding the further enhancement of the system.
Keywords: computer-assisted assessment, concept maps, intelligent assessment system, adaptive assessment

1 Introduction

Computer-assisted assessment of learner's knowledge is not a novel term in computer science and education. The most widespread systems are ones based on various tests where correct answers are pre-defined. The main drawback of such systems is a level of intellectual behavior, which can be assessed. As a rule it is not above the fourth level in the well-known Bloom's taxonomy [4]. Only a few computer-assisted assessment systems, which assess higher levels of intellectual abilities and skills, have been developed. They are based on strongly subject dependent tasks such as essays or free-text responses. These systems use methods of natural language processing and, therefore, are extremely complicated.

The paper offers a reasonable compromise replacing tests by concept maps, which allow to assess higher order skills and simultaneously do not require natural language processing. So, in rather simple systems the idea of adaptive knowledge assessment may be implemented.

The remainder of this paper is organized as follows. Section 2 briefly discusses computer-assisted assessment. Section 3 gives an overview of different assessment tasks based on concept maps. The developed multiagent concept map based intelligent assessment system is described in Section 4. Section 5 introduces computer adaptive assessment. The possibilities to use concept maps in adaptive knowledge assessment using the developed system are discussed in Section 6. Finally, conclusions are presented and some directions for future work are outlined.

2 Computer-assisted assessments

According to [9] the term "computer-assisted assessment" refers to the use of computers in assessment, encompassing delivering, marking and analysis of assignments or examinations, as well as collation and analysis of data gathered from optical mark readers. The most widespread computer-assisted assessment systems are ones based on objective tests [5, 24] that offer a learner a set of questions, answers on which are pre-defined [9]. The mostly used question types are multiple choice questions, multiple response questions, graphical hotspot questions, fill in blanks, text/numerical input questions, etc.

Computer-assisted assessment typically is included in virtual learning environments, e.g. Blackboard (http://www.blackboard.com) or WebCT (http://www.webct.com), or it can be implemented in the form of a specialized assessment system. In the last case there is a set of available tools both from institutional and commercial developers [24]. CASTLE (http://www.le.ac.uk/castle), TRIADS (http://www.derby.ac.uk/assess/newdemo/mainmenu.html), and TAL (http://www.tal.bris.ac.uk) are examples of software developed within the framework of institutional projects. Commercial tools are Hot Potatoes (http://hotpot.uvic.ca), Respondus (http://www.respondus.com), WebQuiz XP (http://eng.smartlite.it/en2/products/webquiz/index.asp), Questionmark™ Perception™ (http://www.questionmark.com/us/home.htm), and others. The analysis of these products allows to define the following functional capabilities of computer-assisted assessment systems: templates for creation of questions, full functionality related to the management of questions (creation, removing, editing, etc.), planning of knowledge assessment activities, defining of feedback, reporting on the performance of both learners and questions, creation and support of question banks, extensive possibilities of question randomizing (different questions at each attempt of a test, different questions for each learner, etc.), various question delivery modes (without returns to already answered questions, re-answering of questions and moving through them, etc.), multimedia integration into questions, and others.

Computer-assisted assessment provides a number of advantages [10, 14, 16, 18, 24]: greater flexibility regarding place and time of assessment, potential for providing assessments for large number of learners efficiently, instant feedback to learners, extensive feedback to teachers, reduced errors in comparison with human marking, decreased time needed for supervising and marking of assessments, and potential for frequent assessments. Besides the advantages computer-assisted assessment systems have also drawbacks [10, 14, 18, 24]: some types of questions cannot be marked automatically as computer-assisted assessment is suited to those questions which require a limited response, unsupervised computer-assisted assessment sessions present a risk of plagiarism and illegal use of other materials, and some learners may have poor skills of information technologies usage.

However, the main drawback of such systems is a level of intellectual behavior, which can be assessed. According to [5, 16], it is not above the fourth level in the well-known Bloom's taxonomy [4], which includes three levels of lower order skills (Knowledge, Comprehension, and Application), and three levels of higher order skills (Analysis, Synthesis, and Evaluation). However, in [9] this assertion is called to be erroneous, but it is pointed out that designing test questions to assess higher order skills can be time consuming and requires skill and creativity.

Tasks such as essays or free-text responses which allow a learner to offer original answers and his/her judgments and assess higher order skills demand more complex structure and functional mechanisms of systems. Such systems are based on artificial intelligence, for example, e-rater [6], c-rater [15], Auto-marking [23], Atenea [20], and others. Unfortunately, essays and free-text responses are strongly subject and language dependent, and, as a consequence, a corresponding assessment system is narrowly focused. From the authors' viewpoint the concept mapping approach offers reasonable balance between requirements to assess higher levels of knowledge and complexity of an assessment system.

3 Concept mapping for knowledge assessment

Concept mapping can be used "to externalize and make explicit the conceptual knowledge that student holds in a knowledge domain" [7]. Concept maps are graphs, which include concepts as nodes and relations between them as arcs. Sometimes so called linking phrases are used. Usually concept maps are represented as a hierarchy with most general concepts at the top of the map and the more specific concepts placed at the lowest levels [17]. Concept maps can have different topologies, too [26].

Assessment based on concept maps can be characterized in terms of: 1) a task that invites a learner to provide evidence bearing on his/her knowledge structure in a domain, 2) a format for the learner's response, and 3) a scoring system to evaluate learner's concept map [21].

In order of issued task and required answer format, it is possible to provide learners with various levels of task difficulties, as well as assess different knowledge levels. One of the ways to deal with different degrees of difficultness is to issue tasks with different degree of directedness [22]. Directedness is connected with information provided to learners. Tasks vary from high-directed to low-directed. High-directed concept map tasks provide learners with concepts, connecting lines, linking phrases, and a map structure. In contrast, in a low-directed concept map task learners are free to decide which concepts and how many of them should be included and how they will be related in their maps. In other words, tasks can be divided in a subset of "fill-in tasks" where learners are provided with a blank structure of a map and lists of concepts and linking phrases, and in a subset of "construct a map tasks" where learners are free to make their choices.

"Fill-in tasks" can be different, too (Table 1). First, they vary on what is provided for learners (concept list, and/or linking phrases list), do they need to define something by themselves or do they need to use linking phrases at all. Second, they vary on how a pre-defined concept map structure is provided: does it contains already some filled concepts and/or linking phrases, or it is empty. "Construct a map tasks" can have the same variety as "fill-in tasks" and also constraints on a number of concepts needed to use in the concept map, and on a structure (should it be strictly hierarchical or have some cycles).

Difficultness degree can be provided also with different number of concepts. Assessments based on "construct a map tasks" more accurately evaluate differences in learners' knowledge structures and elicit more high-order cognitive processes [26].

Table 1. Fill-in tasks

Task	Is provided		Need to define	
	Concepts list	Linking phrases list	Concepts	Linking phrases
A	X	X		
B	X			X
C	X			
D		X	X	
E			X	X
F			X	

4 The intelligent knowledge assessment system

The first prototype of an intelligent assessment system based on concept maps and multiagent paradigm has been developed and tested [2]. Its main purpose is to allow a teacher put into practice the notion of process oriented learning when learners' knowledge is assessed continuously during a learning course. At the moment the system supports only one task: filling of a concept map structure. Two types of links are used. Important conceptual links show that relationships between the corresponding concepts are considered as important knowledge in a given learning course. Less important conceptual links specify desirable knowledge. The linking phrases and direction are not used in the developed prototype.

The system consists of three modules. The administrator module allows to manage data about learners and groups of learners, teachers and learning courses. The teacher's module supports a teacher in the development of concept maps and in examining of learners' final score. The learner's module includes tools for filling of concept maps provided by a teacher and for viewing feedback after his/her solution submission. The modules interact sharing a database which stores data about teachers and their learning courses, learners and groups of learners, teacher-created and learner-completed concept maps, learners' final score and system's users (Fig. 1).

The system's functionality and its client/server architecture are described in details in [2]. Use case diagrams of construction of a new concept map by a teacher and examining of learners' results, as well as concept map filling by a learner are given in [3].

The system supports the following scenario. A teacher divides a learning course into some stages. Using the developed system the teacher prepares concept maps for each stage in the following way. Concepts taught to learners at the first stage are included in the first concept map of the learning course. At the second stage learners acquire new concepts which the teacher adds to

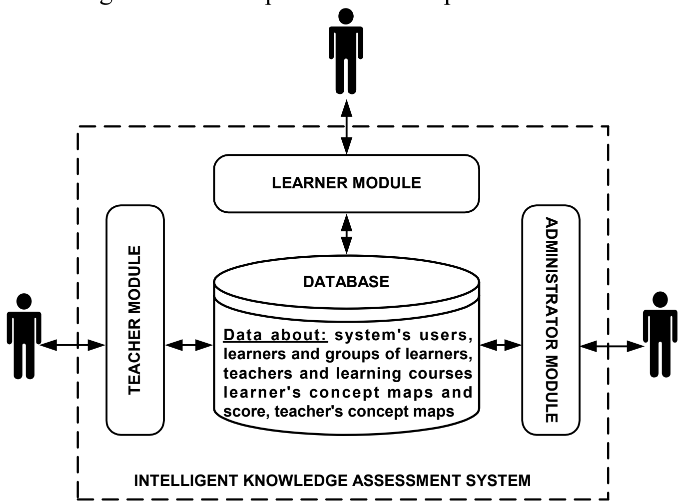


Fig. 1. The architecture of the system

concept map of the first stage without changing the relationships among already existing concepts. Thus, a concept map of each stage is an extension of a concept map of the previous stage. A concept map of the last stage displays all concepts in the learning course and relationships between them. During knowledge assessment learners get a structure of a concept map, which corresponds to the learning stage. At the first stage it is an empty structure with very few initial concepts defined by the teacher. In the subsequent stages new concepts are included in addition with those, which a learner already has correctly inserted during the previous stages. After finishing the concept map, the learner confirms his/her solution and the system compares concept maps of the learner and the teacher on the basis of five patterns described below. The final score and the learner's concept map are stored in the database. The learner receives feedback about correctness of his/her solution. At any time the teacher has an opportunity to examine a concept map completed by the learner and his/her score. Figure 2 displays the described scenario.

The system is a multiagent system, which consists of an intelligent agent for assessment of learners' knowledge level and a group of human agents, i.e. learners who are communicating with this agent. The intelligent assessment agent is a core of the system and its unique feature. It makes the basis of the learner's module and includes the communication, knowledge evaluation, interaction registering, and expert agents described in [2].

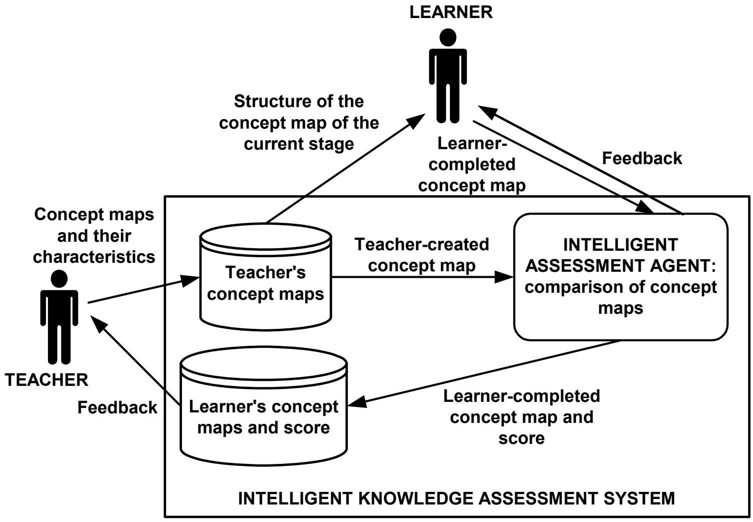


Fig. 2. The scenario of the system's operation

The intelligent assessment agent is responsible for the comparison of teacher's and learner's concept maps using an algorithm sensitive to arrangement of concepts. The main assumption is that the learner's understanding of relationships between concepts has the primary value, while the type of a link and a place of concepts within the structure of a concept map are the secondary things. Five patterns of learner solutions can be recognized by the agent (Fig. 3):

Pattern 1. The learner has related concepts as they are connected within a standard map of the teacher. In this case the learner receives 5 points regarding every important link and 2 points for less important link. Fig. 3b shows that the learner has related concepts A and E which fully matches the same relationship in the teacher-created concept map (Fig. 3a).

Pattern 2. The learner has defined a relationship, which does not exist in a concept map of the teacher. In this case he/she does not receive any points. In Fig. 3c it is shown that the learner has related concepts A and H, but such relationship does not exist in the teacher-created map (Fig. 3a).

Pattern 3. The learner's defined relationship exists in a standard map, the type of a link is correct, but at least one of concepts is placed in an incorrect place. The learner receives 80% from maximum score for that link. Fig. 3d shows that the learner has defined relationship between concepts B and D, which also exists in the teacher's map (Fig. 3a). Both concepts are placed in the incorrect places although the type of the link is correct.

Pattern 4. The learner's defined relationship exists in a standard map, the type of a link is wrong, and at least one of concepts is placed in an incorrect place. The learner receives 50% from maximum score for the correct link. This pattern is displayed in Fig. 3e. Comparing the learner defined relationship between A and F with teacher's one (Fig. 3a) it is easy to see that concept F is placed in an incorrect place, as well as type of the link between concepts is less important instead of important link.

Pattern 5. A concept is placed in a wrong place, but its place is not important. The learner receives maximum score for a corresponding link. Fig. 3f displays that the learner has exchanged concepts M and L by places comparing with the teacher-created concept map (Fig. 3a).

The developed system has been tested in four learning courses and seventy four students were involved (the testing example is described in [2]). The students positively evaluated the chosen approach to knowledge assessment, as well as functionality and user interface of the system (questions and student answers are given in [1]). They also stated desire to use such assessment technique in courses that will follow.

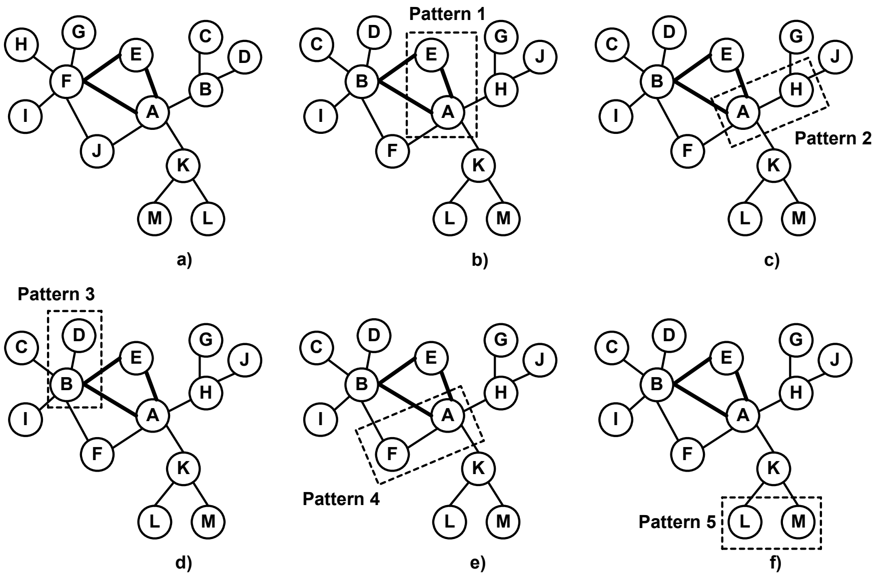


Fig. 3. Patterns for learner's solution evaluation with the intelligent assessment agent: a) a teacher-created concept map; b) – f) patterns within a learner-created concept map

The idea to computerized concept mapping is not new at all. A number of commercial and non-commercial graphical software packages and tools already exist, for example, AXON Idea Processor (web.singnet.com.sg/~axon2000/), Inspiration (www.inspiration.com), IHMC CmapTools (cmap.ihmc.us), which provide such functions as concept map construction, navigation and sharing, and can be used as a useful learning tool. These products do not assess created concept maps. This task can be solved by such tools as COMPASS [12] and the system described in [8]. The developed system has two discriminative features in comparison to them. Both known systems consider assessment as a discrete event, while the system described in this paper supports process oriented learning and allows the teacher to extend the initially created concept map for the new stage of assessment. The second unique feature is an algorithm that compares the teacher's and learner's concept maps and is sensitive to the arrangement and coherence of concepts. The third feature that is under development at the moment is concept map adaptive knowledge assessment described below in this paper.

5 Computer adaptive assessment

Computer adaptive assessment adapts tests to knowledge level of each learner [19], in the following way [25]. A learner receives an assessment item of average difficulty. If he/she does not answer this item or gives an incorrect answer, a less difficult item is presented. Otherwise, he/she gets a more difficult item. This process continues until the predetermined test termination criteria have been met. Therefore, learners with a low knowledge level do not respond to very difficult items, but learners at a high achievement level are not required to answer too simple items. Adaptive assessment provides more accurate conclusions about the actual knowledge level of each learner [19].

This kind of assessment is supported by some software products mentioned in Section 2, e.g., Questionmark™ Perception™ and TRIADS. There is also some research in this area. PASS module [11] is based on adaptive testing and adaptive questions techniques, and can be integrated in an Adaptive Educational Hypermedia System in order to provide personalized assessment selecting the appropriate question for a learner according to the questions' parameters, the assessment parameters, and the current learner's knowledge level. E-TESTER [13] automatically creates questions based on e-learning content provided to a learner.

6 Concept map based adaptive knowledge assessment

Functionality of the developed system can be essentially improved by providing adaptability of offered tasks to knowledge level of a learner. At the moment the same structure of a concept map and initial concepts are presented to all learners irrespective of a level of achievements of a particular learner. The idea of computer adaptive assessment allows to identify at least two possibilities. The first assumes to change the system's behavior when a learner fills the structure of a concept map. The system should monitor each learner's performance during the task. If it determines that the learner has some difficulties, it should intervene by filling some empty places with correct concepts. This process continues until only a few empty places will remain or the learner will complete the task. Despite of simplicity of the approach there are several unsolved issues:

- What methods will allow the system to determine when the learner has met difficulties? Whether the system should take into account how long the learner does nothing before inserting the next concept? Whether the system should track multiple movings of the same concept?
- What principle will the system use to put correct concepts into empty places?
- What number of empty places can serve as a criterion of the termination of system interventions?
- What should the system do with concepts which the learner has incorrectly inserted into structure of a concept map?

The second approach assumes enrichment of the system by tasks of various types. In this case adaptability can be implemented using the following scenario. At the first assessment stage all learners receive the task of the lowest directedness. If the system determines that a learner has some difficulties, it increases a degree of directedness. This process continues until the learner completes the task or reaches the highest degree of directedness. At the next stage the learner receives the task of directedness which he/she has achieved in the previous assessment. The level of task directedness can be increased in case if the learner has successfully completed the task in the previous assessment without lowering the directedness. Of course, it is necessary to store achieved degree of task directedness in a profile of the learner. There are uninvestigated aspects. First of all, it is necessary to find methods which will allow the system to determine that the learner has met some difficulties. The next problem is the large set of concept map based tasks ranging from "fill-in tasks" to "construct a map tasks" (Section 3). All of them cannot be implemented due to huge amount of needed work. Thus, it is necessary to select a subset of tasks. Last, but not least, is the development of the user interface which should offer new tools when the directedness degree of the task will increase.

7 Conclusions and future work

The paper focuses on computer adaptive assessment which main advantage is opportunity to present learners with tasks appropriate to their knowledge level. Concept maps can be successfully used as a core of adaptive assessment systems. Authors believe that usage of concept maps provides possibility to issue tasks with different level of difficultness and to assess fourth and fifth levels of knowledge according with Bloom's taxonomy. At the moment the analysis of the described possibilities concerning adaptive assessment using concept maps is at its early stage.

8 Acknowledgment

This work has been partly supported by the European Social Fund within the National Program "Support for the carrying out doctoral study program's and post-doctoral researches" project "Support for the development of doctoral studies at Riga Technical University".

The main results are outcomes of the research project "Concept maps and ontology based intelligent system for student knowledge self-assessment and process oriented knowledge control".

References

1. Anohina A, Graudina V, Grundspenkis J (2006) Intelligent system for learners' knowledge self-assessment and process oriented knowledge control based on concept maps and ontologies (accepted for Annual Proceedings of Vidzeme University College)
2. Anohina A, Grundspenkis J (2006) Prototype of multiagent knowledge assessment system for support of process oriented learning (accepted for the Proc. of the 7th Int. Baltic Conf. on DB&IS)
3. Anohina A, Stale G, Pozdnyakov D (2006) Intelligent system for student knowledge assessment (accepted for Proceedings of Riga Technical Univ.)
4. Bloom BS (1956) Taxonomy of educational objectives. Handbook I: The cognitive domain. David McKay Co Inc., New York
5. Bull J Introduction to computer-assisted assessment. Available at: http://asp2.wlv.ac.uk/celt/download.asp?fileid=44& detailsid=200008
6. Burstein J, Leacock C, Swartz R (2001) Automated evaluation of essays and short answers. In: Proc. of the 5th Int. Computer Assisted Assessment Conf., pp 41-45
7. Canas A (2003) A summary of literature pertaining to the use of concept mapping techniques and technologies for education and performance support
8. Chang KE, Sung YT, Chen SF (2001) Learning through computer-based concept mapping with scaffolding aid. J. Computer Assisted Learning, vol. 17: 21-33
9. Computer-assisted Assessment (CAA) Centre - http://www.caacentre.ac.uk
10. E-assessment. Available at: http://en.wikipedia.org/wiki/E-assessment

11. Gouli E, Papanikolaou KA, Grigoriadou M (2002) Personalizing assessment in adaptive educational hypermedia systems. In: Proc. of the 2nd Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems. Springer-Verlag, London, UK, pp 153-163
12. Gouli E, Gogoulou A, Papanikolaou K, Grigoriadou M (2004) COMPASS: an adaptive Web-based concept map assessment tool. Proc. of the 1st Int. Conf. on Concept Mapping
13. Guetl C, Dreher H, Williams R (2005) E-TESTER: A computer-based tool for auto-generated question and answer assessment. In: Proc. of the E-Learn 2005-World Conf. on E-Learning in Corporate, Government, Healthcare, and Higher Education, vol. 2005, no. 1, pp 2929-2936
14. Lambert G (2004) What is computer aided assessment and how can I use it in my teaching? (Briefing paper) Canterbury Christ Church University College
15. Leacock C, Chodorow M (2003) C-rater: scoring of short-answer questions. Computers and the Humanities, 37(4): 389-405
16. Mogey N, Watt H (1996) Chapter 10: The use of computers in the assessment of student learning. In: Stoner G (ed) Implementing Learning Technology. Learning Technology Dissemination Initiative, pp 50-57
17. Novak JD, Canas AJ (2006) The theory underlying concept maps and how to construct them. Technical Report IHCM CmapTools 2006-1
18. Oliver A (2000) Computer aided assessment - the pros and cons. Available at: http://www.herts.ac.uk/ltdu/learning/caa_procon.htm
19. Papanastasiou E (2003) Computer-adaptive testing in science education. In: Proc. of the 6th Int. Conf. on Computer Based Learning in Science, pp 965-971
20. Pérez D, Alfonseca E, Rodríguez P (2004) Application of the BLEU method for evaluating free-text answers in an e-learning environment. In: Proc. of the 4th Int. Language Resources and Evaluation Conf
21. Ruiz-Primo MA, Shavelson RJ (1996) Problems and issues in the use of concept maps in science assessment. J. Res. Sci. Teaching 33(6): 569-600
22. Ruiz-Primo MA (2004) Examining concept maps as an assessment tool. In: Proc. of the 1st Conf. in Concept Mapping
23. Sukkarieh JZ, Pulman SG, Raikes N (2003) Auto-marking: using computational linguistics to score short, free text responses. In: Proc. of the 29th Conf. of the Int. Association for Educational Assessment
24. Using computer assisted assessment to support student learning. The Social Policy and Social Work subject centre with the Higher Education Academy (SWAP) – http://www.swap.ac.uk/elearning/develop6.asp
25. What is CAA? (2005) Castle Rock Research Corp.- http://www.castlerockresearch.com/caa/Default.aspx
26. Yin Y, Vanides J, Ruiz-Primo MA, Ayala CC, Shavelson RJ (2005) Comparison of two concept-mapping techniques: implications for scoring, interpretation, and use. J. Res. Sci. Teaching, vol. 42, no. 2: 166-184

Term Clustering and Confidence Measurement in Document Clustering

Kristóf Csorba and István Vajk

Department of Automation Technology and Economics,
kristof@aut.bme.hu, vajk@aut.bme.hu

1 Introduction

Document clustering is the classification of documents into several groups based on a given classification criteria, like the topic similarity. In a supervised learning scenario, the system extracts features from labeled examples and learns to identify documents of the same categories. A large family of methods is based on vector spaces, where documents are represented by vectors in a space of features, like occurrences of the various terms. Every used term (not a stopword and not too rare) is assigned to a feature and the coordinate of the document along this dimension is a function of the occurrence of the term in the documents. A frequently used weighting scheme family is the TFIDF (term-frequency, inverse document frequency) scheme [1].

This paper presents a novel technique for the clustering of documents in scenarios, where there is no strict need to cluster all the documents, and the system is allowed to return some results declared as unsure. This way we achieve a much cleaner result by rejecting the classification of documents with ambiguous topic. The rejection is decided based on a confidence value described in this paper. This has an advantage for the term filtering as well, because some documents are allowed to get ambiguous because of the removal of the majority of its terms.

Our clustering technique uses a vector space model, so we start the procedure by creating the document term matrix (X) with column vectors representing the documents and row vectors standing for the various terms. The values of X are in this case the occurrence numbers of the terms in the given documents.

[4] showed that the roles of documents and terms can be swapped regarding the clustering: we can not only cluster the documents in the space of terms, but the

terms can be clustered in the space of the documents as well. The information bottleneck method aims to perform the document clustering in a more compact feature space. This is achieved by the clustering of the features by preserving as much information for the document clustering as possible. That means features with similar influence on the clustering result are merged.

In this case the information bottleneck method means the employment of a term clustering step before the document clustering, to create term clusters. This way the documents are classified in the space of the term clusters instead of single terms. This is a very strong dimensionality reduction, as only a much smaller number of term clusters in needed for the correct classification, than the number of available individual terms, even after traditional term filtering.

Further parts of this paper are organized as follows: Section 2. describes the novel algorithm for creating term clusters, Section 3. presents how the document clustering works with the term clusters created, and how confidence is measured. Section 4. presents experimental results and finally conclusions are made in Section 5.

2 New term cluster creation method

In the literature the term clustering of the double clustering is usually performed by an unsupervised clustering algorithm [1]. K-means is often used together with the cosine distance measure for this purpose.

Our novel method aims to perform a supervised term clustering together with a strong term filtering. It selects for every target cluster the terms, which allow the best classification of the documents. The term clusters are created iteratively until there is a need for additional ones. The terms not used in any of these term clusters are discarded and not used in the later part of the clustering procedure. Not surprisingly the terms in the first term clusters are the most topic specific terms of the given topics. After every document class has been covered by one term, additional term clusters can be created to increase the number of successfully clustered documents. This is important because all the documents not containing any of these most important terms are mapped into a null—vector in the new feature space and cannot be assigned to any of the document clusters.

The term cluster creation algorithm uses the frequently used clustering performance measures precision, recall and their harmonic average, the F-measure. These are defined as follows: let us suppose we have to select the elements of a document class as good as possible. The result is a document cluster, which contains documents from the given class, and some misclassified ones from other classes. Precision is the amount of correctly selected documents among all selected document. Recall is the rate of correctly selected documents among all documents in the target class. These two measures are usually used together, as none of them is enough on its own: maximal precision can be achieved by selecting one single document from the target class, as there will not be any misclassifications. On the other hand a maximal recall can be achieved by selecting every document. That ensures the selection of every document in the target class, but leads to many misclassified documents as well. For this reason, F-measure is defined as their harmonic middle to enable optimizations based on both criteria.

2.1 The two steps of term cluster creation

The term cluster creation procedure consists of two main steps: the building and the reduction steps. The building step collects the terms which identify the target document class with the highest precision. The reduction step eliminates the redundancies. Overview of the procedure is shown in Fig. 1.

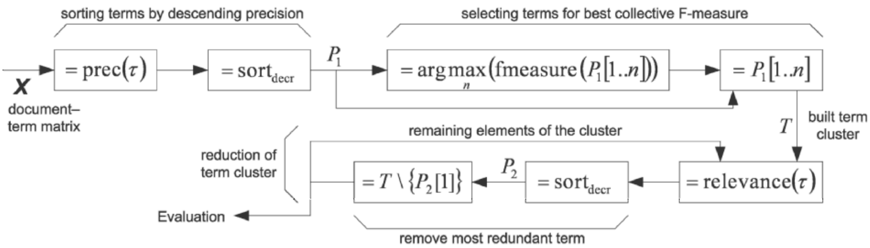


Fig. 1. The term cluster creation procedure.

In the term cluster creation a term is treated to select a document if it occurs in it. Term cluster building for a given document class starts with sorting the terms by descending precision for the selection of the documents of the given class. It then iteratively adds the terms in this order (Fig. 2.)

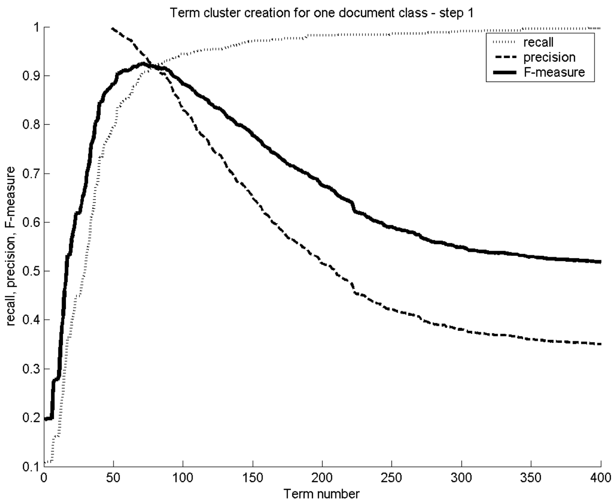


Fig. 2. First step of term cluster creation.

After adding the terms occurring only in the documents of the target class, the precision will begin to decrease. It will still increase the recall of course, because the term cluster will select more and more documents. This leads to a maximum of the F-measure, after which the recall increment cannot compensate the falling precision. Terms are added to the term cluster till this point.

At this point the term cluster may contain terms, which select the same documents. The aim of the second step is to remove such redundancies. The redundancy of a term τ is measured by the change rate of the precision if the term is removed from the term cluster.

$$redundancy(\tau) = \frac{recall(T \setminus \tau)}{recall(\tau)}, \tag{2.1.1}$$

where T is the set of terms in the term cluster.

The redundancies have to be recalculated after every term removal, but as the figure shows, we can expect relative many terms with high redundancy. The second step removes iteratively the terms by selecting always the one with the highest redundancy in a greedy way (Fig. 3.).

As term removal continues, the recall of the remaining term cluster decreases. If it reaches a predefined minimal recall value, the reduction step terminates. The lower the minimal recall limit is, the more terms can be removed from the term cluster.

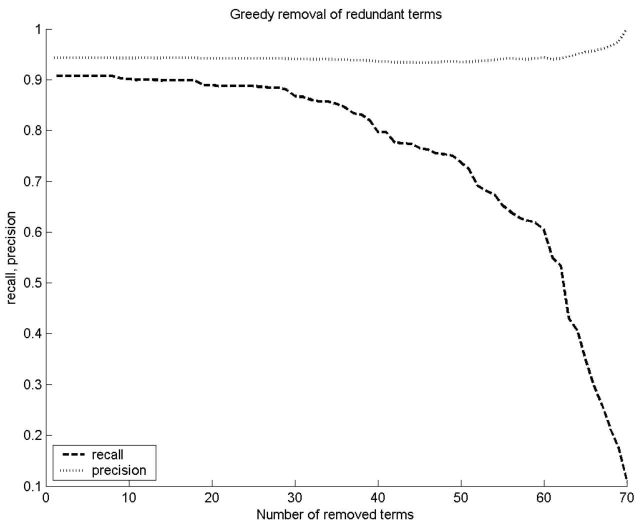


Fig. 3. Second step of term cluster creation.

The term clusters from the document clustering are created after each other for the different target document classes with the same, predefined minimal recall limit. As every term can occur in at least one term cluster, after a cluster has been created, all the terms in it are removed from the set of available terms. Theoretically the term clusters can be created after each other until every term is assigned to a cluster.

2.2 Term clusters for multiple document classes

If we have some document classes with relative near topics, we may expect some terms to occur in more document classes. If a term can select documents from two classes, it can still be useful to separate these two classes from the others. This means we can create term clusters for two document classes as well.

Our method first creates a term cluster for every document class. These are called the basic term clusters. After that the remaining terms are used to build the additional clusters. In every iteration we create a term cluster for every document class and document class pair. Among these temporarily term clusters only the one with the highest F-measure is used as a term cluster in the clustering system.

3 Document clustering and confidence

Once the term clusters are created, they can be used to assign documents to document clusters. Starting from the document term matrix X, a document term-cluster matrix Y has to be calculated by applying the clustering of the terms. This is done by merging of the row vectors in X belonging into the same term cluster into a single row vector in Y:

$$Y_{\theta,\bullet} = \sum_{\tau \in T_\theta} X_{\tau,\bullet}\,, \tag{3.1}$$

where T_θ is the set of terms in the θ-th term cluster. The same operation can be expressed with a matrix multiplication by defining

$$C_{\theta,\tau} = \begin{cases} 1 \, iff \; \tau \in T\theta \\ \quad 0 \; else \end{cases}, \tag{3.2}$$

where τ is the term index. In this case, $Y=CX$. The document clusters can be defined as a linear combination of the term clusters. The matrix Q contains the weights of the term clusters for the document classes similar to Y for the term clustering. This enables the calculation of Z, the basis of the decision as $Z=QY$.

During the supervised learning of the clustering system we have to assign a weight for every (document class; term cluster) pair. The later a term cluster was created, the more good separating terms were already assigned to a previous cluster, which leads to a lower quality term cluster.

$$Q_{\lambda,\theta} = \begin{cases} \dfrac{1}{(\max(d,\theta) - d + 1)^2} & iff\ \lambda \in \Lambda_\theta \\ 0 & else \end{cases} \tag{3.3}$$

where d is the number of document classes and λ_θ is the set of document classes, for which the term cluster with index θ was created. This means that for a given document class λ the Q matrix has nonzero weight for the term clusters, which were created to cover this document class. As the index of the term cluster increases, the quality is expected to decrease and so the weight gets smaller.

3.1 Decision making an its confidence

The matrix Z can be treated to contain scores for every (document; document cluster) pair. Each document is assigned to the document cluster, for which its score is maximal. That means the document δ is assigned to the document class λ for which

$$\lambda(\delta) = \arg\max_{\lambda} \{ Z_{\lambda,\delta} \} \tag{3.1.1}$$

The confidence of this decision can be expressed by the ratio of the maximal and the second maximal score from Z:

$$conf(\delta) = \frac{Z_{\lambda(\delta),\delta}}{\max\limits_{\lambda \neq \lambda(\delta)} \{ Z_{\lambda,\delta} \}} \tag{3.1.2}$$

As the confidence value can be infinite, if only one document class gets nonzero score, we are going to maximize the values to 5 in the following figures.

Fig. 4. and Fig. 5. present the histogram of confidence, if we use only the basic term clusters. If we define a minimal confidence limit, under which we declare the result to be ambiguous, we can avoid some of the false classifications and move them to the unsure category. There are still confident, but false results, which is caused by their infinite confidence. That means, that unfortunately they contain only a few terms used in the term sets, but all of these terms belong to another topic cluster, which leads to a zero score for the correct cluster.

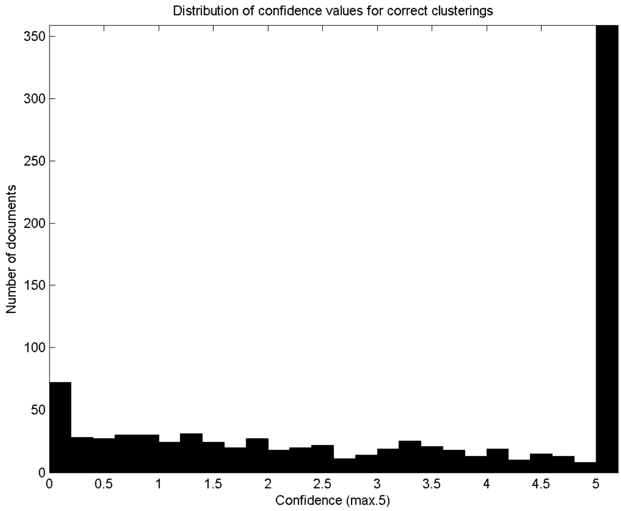


Fig. 4. Example confidence histogram for correctly classified cases.

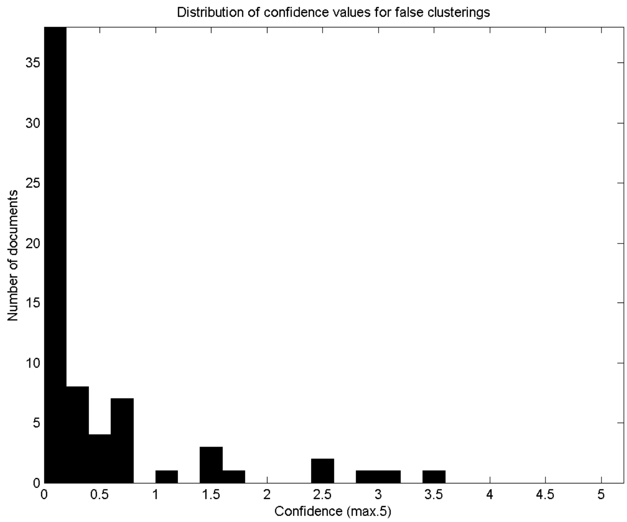


Fig. 5. Example for confidence histogram for misclassified cases.

If there are more term clusters (with even lower separation abilities), the confidences are often lower, but the confidence of the misclassifications is hardly always below 2. That means we can declare results with confidence below 2 to be ambiguous in the current case. This filtering will move most of the false results into the unsure category. The confidence measure shows a behavior one would expect.

3.2 Cascading of classifiers

If we can build a classifier with a training set, which is capable to decide, which results are sure and which are ambiguous, we can easily route the unsure results into a new classifier, which is trained specialized for the ambiguous cases of the previous stage. In this way, we get a cascade of classifiers as shown in Fig. 6.

To train such a cascade the training document set has to be directed into the input of the first classifier. Every document labeled as unsure by this stage can be forwarded into the next classifier. This composition is an effective way for further improvement of the classification quality, because the individual elements can be optimized for maximum precision: the rate of undecided cases gets much less important, which allows to pay less attention to the recall.

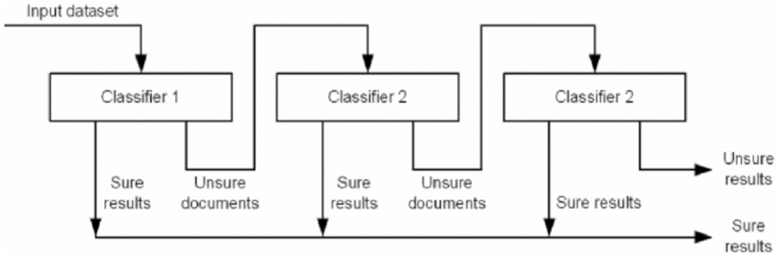


Fig. 6. Cascade of classifiers.

4 Experiments

The experiments presented in the followings were performed on a part of the commonly used dataset 20Newsgroups [2]. The dataset contains news in 20 categories with 1000, usually short documents each. For our evaluation purposes 3 categories were used: “alt.atheism”, “comp.graphics” and “rec.autos”. The set of 3000 documents were divided into a training set with 2000 documents and a testing set with 1000 documents.

First we examined the basic term sets created by the algorithm for the three document clusters. Some of the terms are shown in Table 1. They are definitely important terms of the given topics.

Table 1. Terms in the three basic term clusters

atheism, Christianity, church, Bible, Christians, moral, objective, belief, morality, Christian, religion, notion, beliefs, atheist, murder, context, gods
viewer, TIFF, processing, shareware, polygon, GIF, JPEG, formats, VESA, pixel, VGA, Windows, Mac, pixels, rendering, format, PC
insurance, vehicle, radar, mph, gear, detector, suspension, GM, lights

Using the two minimal recall limits 0.6 and 0.8 we compared the precision of the term clusters the algorithm created. Fig. 7 shows the results. The lower minimal recall limit enables the removal of more terms from the clusters. This increases the precision. Beside this effect, lower limit removes more terms which leads to more possible term clusters from the given set of all terms.

Using only the first t term clusters we can control the number of used terms. As this increases, precision increases slightly, because the few misclassified documents are moved into the category of ambiguous results. The increment in performance is shown in Fig. 8. The number of terms begins with the total term number of the base term clusters, as before creating these, the overall performance of the system cannot be measured.

Although the increment of the precision is very useful, it leads to a strong decrement in the rate of sure results, because using more terms to calculate scores decreases the confidence values.

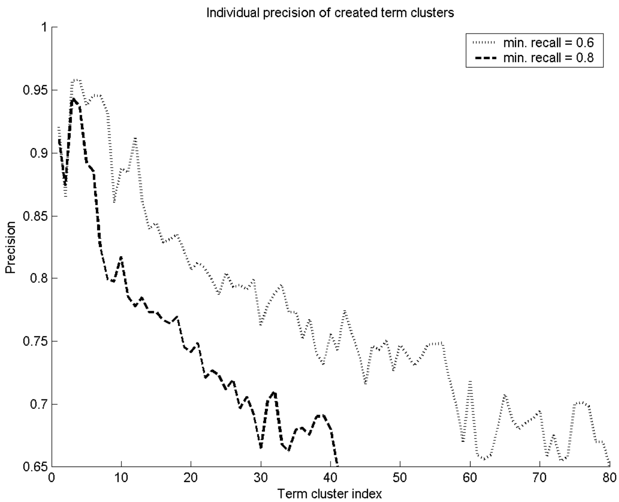


Fig. 7. Precision of the created term clusters for various minimal recall limits.

Using classification units created with high minimal recall and confidence limits, the performance of a cascade was analyzed. Fig. 9. shows the number of correctly, false and unsure assigned documents. The number of correct classifications increases clearly in the beginning, but more than 3 elements in the cascade seem not to be useful. Most of the unsure documents in the beginning are classified correctly by the second and third stages and only a few documents are assigned to the false topic.

A disadvantage of the cascade is the increasing number of used terms, as the second and third stages in the measurement above use nearly 400 terms together, although the first element used only 60. The reason for the increasing term need is the fact, that the documents reaching the second stage cannot be classified with the most frequent high-precision terms anymore. Rarer terms need to be used, from which the system needs more for a confident classification.

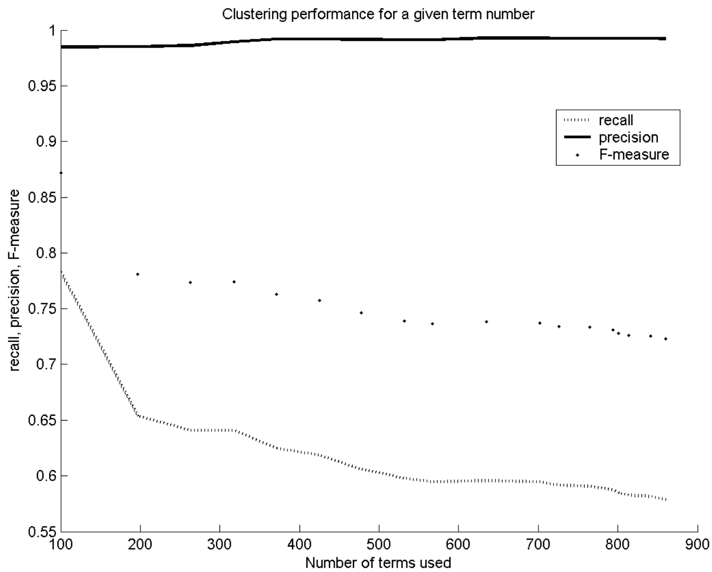


Fig. 8. Clustering performance with a given term number.

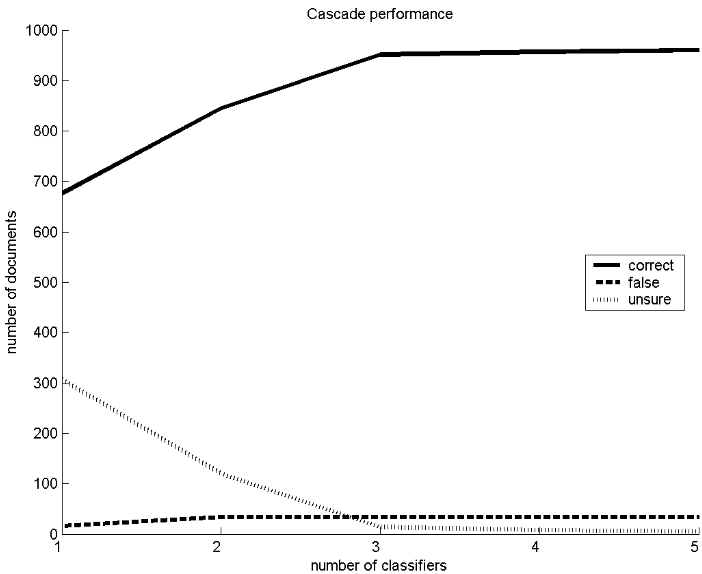


Fig. 9. Performance of a cascade of classifiers.

5 Conclusions

We have presented a novel term clustering and term filtering technique, which aims to cover the document classes in a greedy way by selecting the terms with the highest precision and after achieving the maximal F-measure, it removes redundant terms to minimize the term cluster size. Combined with the confidence measurement in the document clustering procedure, an effective way was shown for the document clustering in situations, where the entire document set does not have to be clustered, because a 30-40% ambiguous result is acceptable. This is often the case in text retrieval systems, where as much as possible documents have to be retrieved, but discarding of the ambiguous documents is still better than risking the false classification.

If the current application does not allow discarding the unsure documents, the base technique is still applicable by cascading multiple classifiers. Every element of a cascade becomes the unsure results of the previous stage as input. This way every cascade element will be specialized to different document types. This allows a high precision clustering, but the more elements we use in a cascade, the more terms are needed for the whole procedure.

6 Acknowledgments

This work has been fund of the Hungarian Academy of Sciences for control research and the Hungarian National Research Fund (grant number T042741).

References

1. Singhal (2001) Modern information retrieval: A brief overview. In: IEEE Data Engineering Bulletin, vol. 24, no. 4, pp. 35–43.
2. K. Lang (1995) Newsweeder: Learning to filter netnews. In: ICML, pp. 331–339.
3. L. Li and W. Chou (2002) Improving latent semantic indexing based classifier with information gain. In: tech. rep., May 16 2002.
4. N. Slonim and N. Tishby (2000) Document clustering using word clusters via the information bottleneck method. In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Clustering, pp. 208–215.

Automation of the Ontology Axioms Transformation into Information Processing Rules Model

Olegas Vasilecas, Diana Bugaite

Department of Information Systems, Vilnius Gediminas Technical University, Lithuania. Olegas.Vasilecas@fm.vtu.lt, diana@isl.vtu.lt

1 Introduction

The business rule approach is used in information systems to represent domain knowledge and to maintain rules systems efficiently in volatile business environment. But firstly, it is necessary to determine the business rules and ensure that the rules are appropriate. The important requirement for developing of the rules model is to reduce efforts, costs and time. This requirement can be implemented by the explicit use of enterprise knowledge for generation of rules, which can be implemented as modern DBMSs triggers.

In computer science, ontologies are used to represent real-world domain knowledge. Therefore, knowledge represented by ontology can be used for generation of rules. Moreover, ontology expressed in a formal way [1] can be transformed into rules model automatically.

In this paper, a method for representing knowledge by ontology transformation into the rule model is presented. The method is based on transforming ontology axioms into information processing rules.

2 Related Work

A definition of a business rule (BR) depends on the context in which it is used. From the business system perspective, a BR is a statement that defines or constrains some aspects of a particular business. At the business system level, BRs are expressed in a declarative manner. [2].

From the perspective of information systems (IS), a BR is a statement, which defines the major rules of information processing using a rule-based language [3].

At the execution level (or software systems level), rules are statements that are transferred to the executable rules, like active DBMS triggers.

Information-processing rules are used in ISs to process the required information correctly. These information-processing rules are derived from BRs, which are taken from the business system level. In practice, information-processing rules are implemented by executable rules. Information-processing rules should be expressed as ECA (event-condition-action) rules to be implemented by executable rules, like active DBMS SQL triggers.

Therefore it is necessary to determine and elicit rules from the application domain and develop ECA rules.

One of the possible ways to solve the defined problems is the use of the domain ontology.

The term 'ontology' is borrowed from philosophy, where Ontology means a systematic account of Existence. In computer science, the definition of ontology is rather confusing. By [4] all definitions of the term 'ontology' attempt to explain what an ontology is from three different aspects: the content of an ontology, the form of an ontology and the purpose of an ontology.

Gruber defined ontology as a specification of a conceptualisation [5]. This definition explains the content of ontology, but it is confusing because it does not explain what a conceptualisation is. According to Genesereth, a conceptualisation includes the objects and their relations which an agent presumes to exist in the world. The process of a conceptualisation is the process of mapping an object or a relation in the world to a representation in our mind. [4]

Ontology defines the basic terms and their relationships comprising the vocabulary of an application domain and the axioms for constraining the relationships among terms [2]. This definition explains what an ontology looks like [4].

In the simplest case, an ontology describes a hierarchy of concepts related by particular relationships (like, is-a, part-of, e.g.). In more sophisticated cases, constraints are added to restrict the value space of concepts and relationships. They, for example, express cardinality, possible length (like, maxLength, minLength...)... In most sophisticated cases, suitable axioms are added in order to express complex relationships between concepts and to constrain their intended interpretation [1].

Ontologies are being built today for many reasons. The reason of creating an ontology depends on research field and an application area where it is going to be used. In this paper, ontology is used for its transformation into the rules model.

3 On Ontology Axioms Transformation into Business Rules

In the application domain or ontology, to which the BRs belong, they are not always expressed in terms of ECA rules. Some of these BRs have explicit or implicit condition and action parts. The missing condition can always be substituted with a default condition state as TRUE. Some BRs may have no explicit action since they can state what kind of transition from one data state to another is not admissible. [6]. But the majority of these BRs do not define explicitly or implicitly the event. There are three possible ways to trigger rules: automatically trigger all rules every time when any related event occurs, trigger rules manually when somebody decides it is necessary, specify necessary events and link them to actual rules. In this research, the third way was used for rules triggering, since the specification of the events and their linking to actual rules enable the system automatically react to the defined events and perform the defined operations, e.g. trigger rules automatically. Moreover, it is not necessary to execute all rules when some event occurs. Only related rules are executed.

Obviously, it is confusing to form information-processing rules of ECA form and consequently implement them by executable rules.

Since BRs are captured in ontology by axioms and constraints of relationships among terms [2], ontology axioms (and ontology as a whole) represented in a formal way can be transformed into BRs (and into conceptual schema) automatically. Moreover, it facilitates BRs transformation into consequent information-processing and executable rules.

The general schema of axioms transformation into BRs is presented in the Fig. 1. It is independent of implementation.

Axioms don't stand alone in ontology. Since axioms define constraints on terms, terms are used to specify axioms. Therefore, those terms and their relationships should be transformed into (conceptual) schema in parallel with transformation of axioms into BRs. For the sake of simplicity ontology axioms transformation into BRs is analysed only.

Axioms define the state in which the domain should be. E.g., axioms can have clearly defined action and, sometimes, condition. Events are not defined in axioms. Therefore, it is necessary to develop some rules to transform ontology axioms and events into ECA rules.

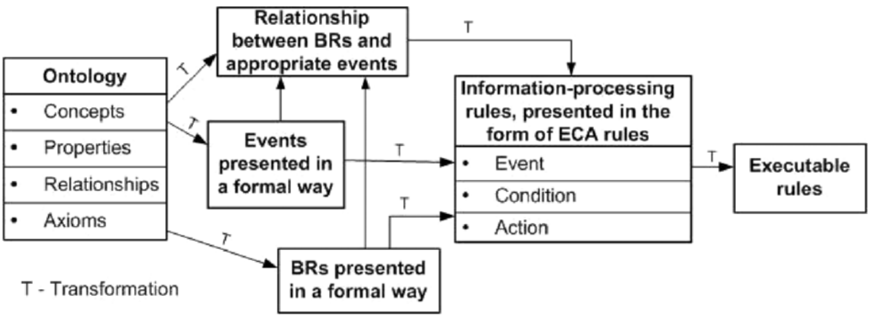


Fig. 1. Transformation of ontology axioms into ECA rules and consequent executable rules.

By [7, 8, 9, 10, 11] events are defined by terms used in ontology vocabulary.

Terms can be used to link axioms with appropriate events, since ontology axioms and events are defined by terms used in ontology vocabulary.

4 A Case Study of Ontology Axioms Transformation into Rules Model

The ontology for a particular business enterprise was created using Protégé-2000 ontology development tool to support the statement of the authors that ontology axioms and events can be transformed into information-processing rules and consequent executable rules. We chose Protégé-2000 to develop the ontology because it allows the open source software to be installed locally. A free version of the software provides all features and capabilities required for the present research as well as being user-friendly. [12]

The axioms are implemented in Protégé-2000 ontology by the Protégé Axiom Language (PAL) constraints. PAL is a superset of the first-order logic which is used for writing strong logical constraints [13].

PAL provides a set of special-purpose frames to hold the constraints. PAL constraints are presented as instances of the *:PAL-CONSTRAINT* class. The class has the following slots [14]:

- *:PAL-name*, which holds a label for the constraint;
- *:PAL-documentation*, which holds a natural language description of the constraint;
- *:PAL-range*, which holds the definition of local and global variables that appear in the statement;
- *:PAL-statement*, which holds the sentence of the constraint.

The main part of the PAL constraint is the *PAL-statement*, which can be mapped to the BR and consequently to the ECA rule.

The EZPal Tab plug-in is used to facilitate acquisition of PAL constraints without having to understand the language itself. Using a library of templates based on reusable patterns of previously encoded axioms, the interface allows users to compose constraints using a “fill-in-the-blanks” approach. [15].

Examples of definition of axiom using the EZPal Tab are the following:

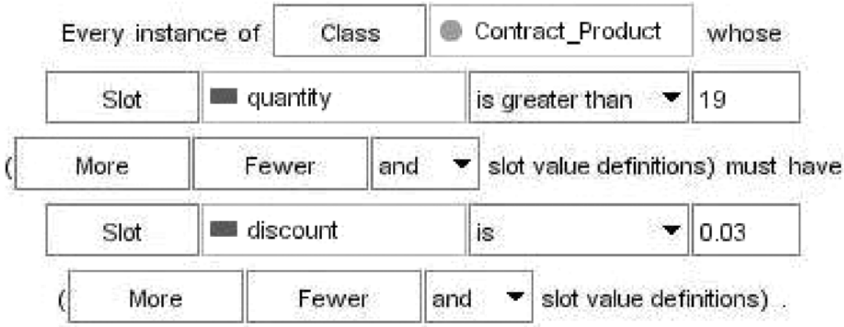


Fig. 2. An example of axiom creation using the EZPal templates.

The non-formal interpretation of the axiom is:

The discount of a contract product depends on quantity of units of a contract product customer buys per time.
If quantity is greater then 19, discount is 3 %.

Fig. 3. An example of axiom creation using the EZPal templates.

The non-formal interpretation of the axiom is:

Every Contract must have a Contract_Product.

PAL constraints and SQL triggers were analysed in details to automate constraints transformation into SQL triggers. The schema of PAL constraints transformation into SQL triggers is presented in Fig. 4.

The main parts of PAL constraints (denoted by grey clouds) are transformed into the main parts of SQL triggers (denoted by grey clouds). ‘PAL-documentation’, ‘PAL-name’ and ‘PAL-range’ are transformed into SQL ‘Comment’, ‘trigger_name’, ‘table | view’ without significant changes. ‘if statement or condition’ and ‘action or possible state’ are transformed into ‘sql_statement’.

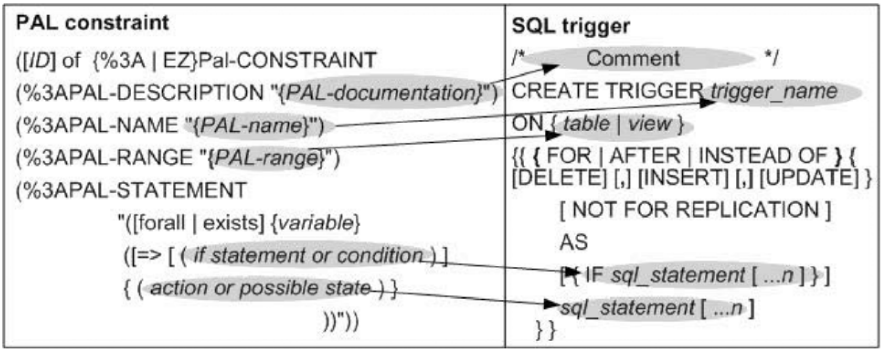


Fig. 4. The detailed schema of automatic transformation of ontology axioms into SQL triggers.

An event of a trigger (DELETE, INSERT, UPDATE) should be taken from defined events in ontology.

The relationships between axioms and events are determined by classes, used to define particular axioms and events. An example of the event is the following:

Fig. 5. An example of the event 'Making contract'.

The event 'Making contract' (Fig. 5) and the axiom 'contract-must-have-contract-product' (Fig. 3) use the same class 'Contract' in their definitions. Moreover, the axiom 'contract-must-have-contract-product' is directly related with the axiom 'discount-3-percent' (Fig. 2). Therefore, the following ECA rule can be generated:

When '*Making contract*', if '*quantity*' of a '*Contract_product*' '*is greater than 19*', then '*discount*' of a '*Contract_product*' '*is 3 %*'.

The plug-in for automatic transformation of constraints is under development now. The consequent plug-in for ontology transformation into the conceptual model is described in [16], since both axioms don't stand alone in ontology and constraints don't stand alone in conceptual model.

The next step of the research is to develop the necessary plug-in and full case study employing the proposed concepts and ideas of the proposed method.

5 Conclusions and Future Work

The analysis of the related works on knowledge-based information system development using the domain ontology shows that the business rules are part of knowledge represented by the ontology. Business rules are captured in ontology by axioms and relationships-constraints of the terms.

The method for ontology transformation into business rules, which are implemented by information-processing rules, was offered. We argue that the ontology axioms can be used to create a set of information-processing rules. They can be transformed into ECA rules and then to active DBMS triggers. Such transformation is possible, since ontology axioms can be mapped into active DBMS SQL triggers.

The experiment shows that the suggested approach can be used to transform ontology axioms described in a formal way into SQL triggers. For this transformation, a suitable tool – Protégé-2000 – was chosen.

PAL constraints can be transformed into ECA rules only manually at the moment. The plug-in for automatic transformation of constraints is under development now.

The next step in our research should be developing prototype and the refinement of the suggested method.

References

1. Guarino N (1998) Formal Ontology and Information Systems. In: FOIS'98, Trento, Italy, 1998. Amsterdam, IOS Press, pp 3-15
2. Bugaite D, Vasilecas O (2006) Ontology-Based Elicitation of Business Rules. In: Nilsson AG et al. (eds.) Advances in Information Systems Development, Springer, Sweden, pp 795-806
3. Lebedys E, Vasilecas O (2004) Analysis of business rules modelling languages. In: IT'2004, Lithuania, Technologija, pp 487-494
4. Yang X (2001) Ontologies and How to Build Them. (March, 2006): http://www.ics.uci.edu/~xwy/publications/area-exam.ps
5. Gruber T (2004) What is an Ontology? (March, 2004): http://www-ksl.stanford.edu/kst/what-is-an-ontology.html
6. Valatkaite I, Vasilecas O (2004) On Business Rules Approach to the Information Systems Development. In: Linger H et al. (eds.) Proc. of ISD'2003, Australia, Kluwer Academic/Plenum Publishers, pp 199-208
7. Cilia M, Bornhovd C, Buchmann AP (2005) Event Handling for the Universal Enterprise. Information Technology and Management - Special Issue on Universal Enterprise Integration, 5(1): 123-148
8. Chen H, Perich F, et al. (2004) SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. In: Proc. of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and

Services, Boston, (March, 2006): http://ebiquity.umbc.edu/get/a/publication/105.pdf.
9. Event Ontology. KendraBase - server 3 (March, 2006): http://base4.kendra.org.uk/event_ontology.
10. eBiquity Event Ontology :: Classes and Properties. SchenaWeb (March, 2006): http://www.schemaweb.info/schema/SchemaInfo.aspx?id=114.
11. Chen H (2004) SOUPA Ontology 2004-06, Semantic web in ubicomp, (March, 2006): http://pervasive.semanticweb.org/soupa-2004-06.html.
12. Jakkilinki R, Sharda N, Georgievski M (2005) Developing an Ontology for Teaching Multimedia Design and Planning. (September, 2005): http://sci.vu.edu.au/~nalin/MUDPYOntologyPreprintV2.pdf.
13. Grosso W (2002) The Protégé Axiom Language and Toolset ("PAL"). (September, 2005): http://protege.stanford.edu/plugins/paltabs/pal-documentation/index.html.
14. PAL Conceptual Framework (2005) (February, 2005): http://protege.stanford.edu/plugins/paltabs/pal-documentation/lang_framework.html.
15. Hou J (2005) EZPal Tab, Stanford University (March, 2006): http://protege.stanford.edu/plugins/ezpal/.
16. Vasilecas O, Bugaite D, Trinkunas J (appear) (2006) On Approach for Enterprise Ontology Transformation into Conceptual Model. In: Rachev B, Smirkarov A (eds.): Proc. of the International Conference on Computer Systems and Technologies "CompSysTech'06", Varna, Bulgaria, pp 281-289

Using Common Process Patterns for Semantic Web Service Composition

Xiaofeng Du, William Song, Malcolm Munro

Durham University, South Road, Durham, DH1 3LE, UK, {xiaofeng.du, w.w.song, malcolm.munro}@durham.ac.uk

Abstract: The web service composition has become an activity research area to improve the usability of web services. By composing web services, enterprise can rapidly deliver cheaper and better business services to their customers. However, how to efficiently select and compose required services is a challenge. The challenge is caused by the gap between customer requirement and service description because the current service description technology lacks abstraction and has no semantics. Many research works on service composition are integrating semantics to solve the service description problem, but they have not sufficiently addressed abstraction. In this paper we propose an approach, named common process pattern, which can improve the efficiency of service discovery and composition. A common process pattern instance abstractly describes a set of services and their relationships based on a commonly used business process. Because of the abstraction provided by this approach, a common process pattern instance's description can be much closer to a customer requirement than a technical service description. Therefore, the service discovery becomes easier.

1 Introduction

Web service composition has become a major research area [3] whose outcome can support efficient and effective business integration so that the enterprises can rapidly deliver better business services to their customers with low cost. How the existing services can be efficiently selected and composed to fulfil the customer requests is a challenge. One of the aspects causing the difficulty is the gap between customer requirement and service description. A customer requirement is normally abstract and business oriented, but a description of a service is concrete and technology oriented. The customers considered here are general business customers who have little or no technical knowledge about the actual services, e.g. parameters type and communication protocols described in a WSDL [2] file. So it is very unlikely to get a customer's request like "*call an air ticket booking service* *with input parameter: String 'airport code', integer 'number of ticket', String 'outgoing date', and String 'return date' and followed by a hotel booking service with input parameter: String 'location', integer 'number of people', String 'start date', and integer 'number of days'*". Rather they may state the request like "*I want to go to New York on 1st of March for a week*". Customers are only interested in the abstract business services which the company can provide for them, not the concrete services used to achieve these business services. Therefore, a mapping from the abstract business services to the concrete services is crucial for efficient service composition and fulfilling customer requirements with low cost. Martin et al. [8] introduced a semantic language, named OWL-S, to add a semantic layer on the top of WSDL service description and provide syntax to describe high level composite processes. However, the OWL-S is still insufficient to bridge the gap because it does not address enough abstraction in the service description. In this paper we propose a pattern based approach to overcome the abstraction problem of service description in order to bridge the gap between the customer requirement and concrete service description.

The main contributions of this paper are the following:

- Introduces a common process pattern approach.
- Builds up a stable abstract description layer on the top of existing services in order to solve the service availability problem in the open Internet service pool.
- Extends the OWL-S language to describe abstract services and common process pattern and implements a prototype to implement the common process pattern.

The rest of the paper is organized as follows. Section 2 discusses the generic approach to service composition and requirement decomposition. Section 3 presents the common process pattern approach together with its definition and description language, and how it can fit into the business model. Section 4 discusses how to implement the common process pattern by using our prototype. Some related work is discussed in section 5. Finally, Section 6 presents the conclusions and future work.

2 Generic Approach

The generic approach (Top-down approach) for mapping the customer request to the concrete services uses knowledge decomposition process which is similar as people solving a problem using their existing knowledge. A customer requirement can be decomposed into several tasks and each task can be further decomposed into subtasks until each of the subtasks is primitive task [5] which can be possibly accomplished by an existing service.

Once the requirement decomposition is done, the business processes rules and AI planning techniques [5] [13] can be applied to form those tasks into a workflow. The dataflow among tasks is based on the data dependency of the required services and the control flow is based on the business processes rules. The services for achieving the tasks in the workflow are located dynamically by using the semantic matching methods [9] based on the specified requirements and goals of each task. Figure 1 illustrates a request decomposition process and a possible workflow after the request decomposition. The dashed lines represent the data flow among the concrete services.

From previous discussion we can summarise some problems exposed by the top-down approach.

- After the request decomposition, the requirements for each task are specified so implicitly that the automatic service discovery is very hard to achieve.
- Due to the uncertainty of customer requirements, it is not guaranteed that the required services can be found because such a required service may not exist at all.
- The workflow formation at runtime is difficult, so is the semantics of input/output parameters to be determined at run time [7]. For example, an address has been provided by a directory service, but it is very confused for a delivery service to decide whether this address is a "from address" or "to address" at runtime. One solution could be that all kind of addresses are explicitly specified in an ontology, and then this ontology will be extremely large.
- Some common requirements and widely used non-primitive tasks have to be repeatedly decomposed into primitive tasks.

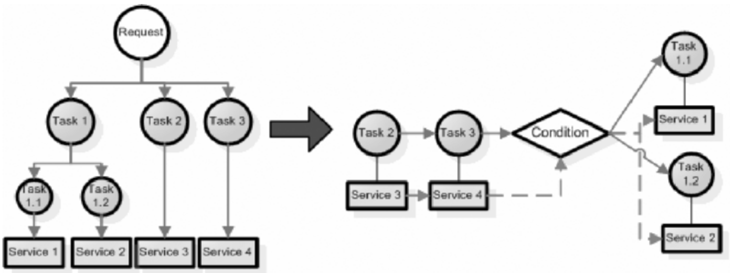


Fig. 1. Request decomposition process and a possible workflow.

3 Common Process Pattern

To address the problems discussed in the previous section, this paper introduces an approach which is a combination of the generic top-down approach with a bottom-up approach, termed Common Process Pattern (CPP). A CPP instance is an abstract representation of a solution for a common business process. It contains abstract specifications to explicitly specify how the business process can be performed and what kinds of services are required. It is generated using the bottom-up approach which is started with existing web services rather than customer requirements. The existing services can be composed together based on the service properties analysis and semantic relationships between services. Many composite services can be generated from a large service pool, but some of them may not be useful or meaningful. Then the top-down approach can be applied here to evaluate whether the new composite services are useful for fulfilling customer requirements. If a composite service can achieve a customer requirement or a non-primitive task, it can be considered as useful because thereafter the requirement or non-primitive task does not need to be decomposed anymore. Then the useful composite service can be abstracted into a CPP instance, which can be described using an extended OWL-S semantic language, for future reuse. Figure 2 illustrates the CPP instance generation process.

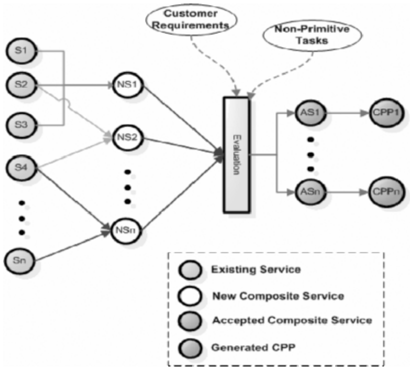


Fig. 2. CPP Generation Process, combination of top-down and bottom-up approaches.

3.1 Motivation and context

A CPP instance can be considered as an abstract composite service. The difference between a CPP instance and a composite service is that the CPP instance only defines how the services can be composed together, i.e. the workflow structure including control flow and dataflow, and the explicit specifications of required services. It does not link to any concrete service. When a CPP instance is executed, the required services are dynamically linked. This technique can prevent the risks brought about by the unavailability of hard coded services in a composite service. Because the CPP instance is generated using bottom-up approach, it is guaranteed that each CPP instance is executable because all the required services exist. Although some existing services may be not available temporarily at runtime, other services with the same type can be found and used as replacement if we assume the service pool is large enough. The usability of the CPP instance is also guaranteed because it has been evaluated by current customer requirements or non-primitive tasks using the top-down approach.

In general a pattern is "the abstraction from a concrete form which keeps recurring in specific non-arbitrary contexts" [11]. The key characteristics of a pattern are abstraction and recurrence. Although this definition is used to define the pattern for object-oriented design, it is still suitable to define the characteristics of other kinds of pattern. Refer to the CPP approach introduced in this paper, a CPP instance can represent a solution for recurring business processes and give the abstract description of the solution and requirement for this recurring business process. An image of a CPP instance is illustrated in figure 3. The circle holes in the diagram represent the missing service that need to be filled in. The thin dashed lines represent the data flow between services and the control structure. The clouds represent the requirements specification for each required service. Because a CPP instance is defined as an abstract composite service, see section 3.2, it can act as a building block to build up other CPP instances. This is also a key feature of CPP approach because it enables CPP instances to build up a hierarchical structure in order to fit for different abstraction levels.

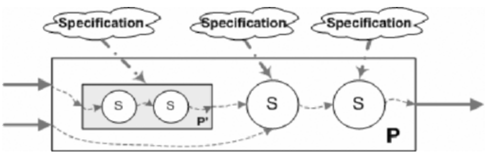


Fig. 3. An Image of the CPP Approach.

A four layers business model proposed by Song [15] shows how an enterprise is operated, see figure 4. The CPP approach can be nicely fitted into this model. It can help to ease the construction of the workflow. A CPP instance describes how a complex task can be achieved by existing resources. Thereafter, that complex task does not need to be further decomposed anymore. The CPP approach also can bring concrete web service descriptions much closer to customer requirements by building up a hierarchical structure of CPP instances to fit for different levels of business requirements.

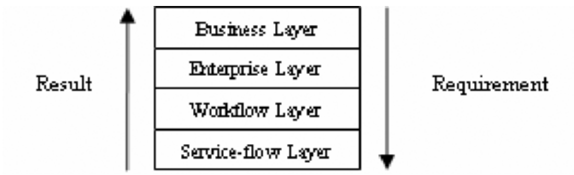


Fig. 4. Four layers business model, the arrows represent the directions of the requirement and the result.

3.2 Formal definition

Before the definition for a CPP instance is given, some basic definitions for concrete web service and abstract web service are given.

Definition 1: Based on the model introduced by OWL-S [8], a concrete service can be represented as: $CS = <Des, I, O, preCon, postCon, T, Grding>$

Where:

Des: A description of the service.

$I = \{I_1, I_2, ..., I_n\}$: A set of inputs.

$O = \{O_1, O_2, ..., O_n\}$: A set of outputs.

$preCon = \{preCon_1, preCon_2, ..., preCon_n\}$: A set of pre-conditions.

$postCon = \{postCon_1, postCon_2, ..., postCon_n\}$: A set of post-conditions.

T: The type of the service.

$Grding$: The service grounding to bind the service with WSDL.

An abstract service can be considered as a specification for a set of concrete services. If a concrete service appears in this set, then this concrete service is a realization of an abstract service [10]. The service grounding is not applicable to an abstract service because an abstract service cannot be invoked directly.

Definition 2: An abstract service AS can be represented as:
$AS = <Des,\ IS,\ OS,\ preCon,\ postCon,\ TS>$

Where:

Des: A description of the abstract service.

$IS = \{IS_1, IS_2, ..., IS_n\}$: A set of specifications of inputs.

$OS = \{OS_1, OS_2, ..., OS_n\}$: A set of specifications of outputs.

$preCon = \{preCon_1, preCon_2, ..., preCon_n\}$: A set of pre-conditions.

$postCon = \{postCon_1, postCon_2, ..., postCon_n\}$: A set of post-conditions.

TS: A specification of the service type.

A CPP instance can be considered as an abstract composite service and its definition is similar to the composite service definition, but some modifications have to be made in order to represent the abstraction.

Definition 3: A composite service can be defined as:
$CptS = <Des, I, O, preCon, postCon, CS, T, DS, CtrlS>$

Where:

Des: A description of the composite service.

$I = \{I_1, I_2, ..., I_n\}$: A set of inputs.

$O = \{O_1, O_2, ..., O_n\}$: A set of outputs.

$preCon = \{preCon_1, preCon_2, ..., preCon_n\}$: A set of preconditions.

$postCon = \{postCon_1, postCon_2, ..., postCon_n\}$: A set of post-conditions.

$CS = \{CS_1, CS_2, ..., CS_n\}$: A set of concrete services used for composing the composite service.

T: The type of the composite service.

$DS = \{(o,i) \mid o \in CS.O \wedge i \in CS.I\}$: DS represents the data flow which is a set of arcs, where each arc connects one service's output to the other service's input. $CS.I$ and $CS.O$ are sets of inputs and outputs of CS.

$CtrlS = < stmt, cdt >$: $CtrlS$ represents the control structure, where $stmt$ is a set of control statements and cdt is a set of conditions.

Definition 4: A CPP instance (an abstract composite service) can be defined as:
$P = < Des, I, O, preCon, postCon, AS, T, DS, CtrlS >$

Where:

Des: A description of the composite service.

$I = \{I_1, I_2, ..., I_n\}$: A set of inputs.

$O = \{O_1, O_2, ..., O_n\}$: A set of outputs.

$preCon = \{preCon_1, preCon_2, ..., preCon_n\}$: A set of preconditions.

$postCon = \{postCon_1, postCon_2, ..., postCon_n\}$: A set of post-conditions.

$AS = \{AS_1, AS_2, ..., AS_n\}$: A set of abstract services.

T: The type of the pattern.

$DS = \{(o,i) \mid o \in AS.O \wedge i \in AS.I\}$: DS represents the data flow which is a set of arcs, where each arc connects one service's output to the other service's input. $AS.I$ and $AS.O$ are sets of inputs and outputs of AS.

$CtrlS = < stmt, cdt >$: $CtrlS$ represents the control structure, where $stmt$ is a set of control statements and cdt is a set of conditions.

The above definition illustrates that the difference between a composite service and a CPP instance is that the CPP instance only defines a set of abstract services rather than a set of concrete services in order to achieve dynamic service discovery and invocation at runtime.

To describe how the inputs and outputs of a CPP instance are related to its internal services, a directed partial graph G_p (V_p, E_p) is used to represent the internal structure of a CPP instance and the connections with outside environment, see figure 5, where

1. V_p: is a set of internal abstract services.
2. E_p: is a set of arcs indicating the data relationships between internal services and internal and external services.
3. $V_p \subset V \wedge E_p \subseteq E$, where there is a whole directed graph G (V, E)

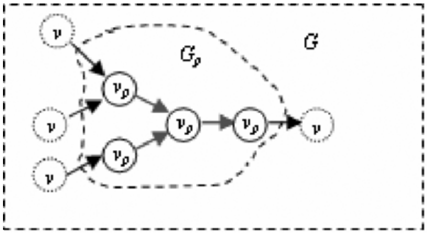


Fig. 5. Partial Directed Graph, black arcs represent inputs and outputs.

The inputs of a CPP instance can be represented as P_{input} which is a set of arcs that satisfy $P_{input} = \{(v, v_p) \mid v \notin V_p \wedge v_p \in V_p \wedge (v, v_p) \in E_p\}$

The outputs of a CPP instance can be represented as P_{output} which is a set of arcs that satisfy $P_{output} = \{(v_p, v) \mid v \notin V_p \wedge v_p \in V_p \wedge (v_p, v) \in E_p\}$

3.3 Description

As discussed previously, a CPP instance is an abstract composite service, thus a semantic language which can address composite service is required to describe a CPP instance. In order to properly describe a CPP instance, we propose a language which extends OWL-S. The upper ontology of OWL-S [8] has been extended to address the abstract composite services and patterns, see figure 6. In the following paragraphs we show how to use the extended OWL-S to describe a CPP instance.

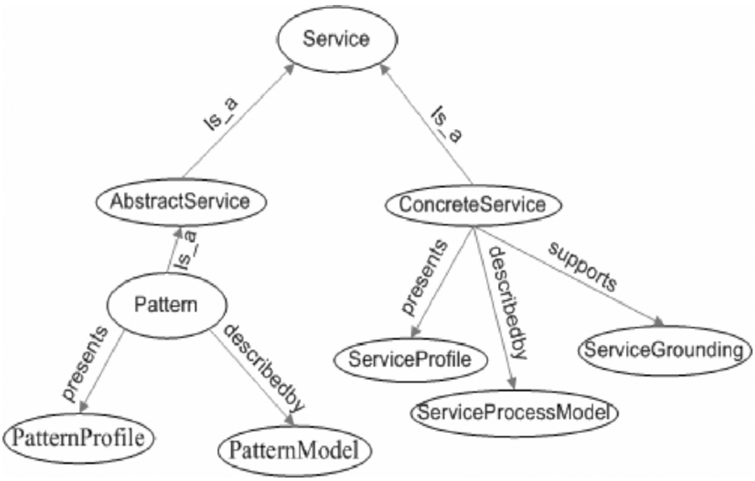


Fig. 6. Extended OWL-S upper ontology.

Suppose we have a CPP instance for the task: calculating the volume of a cylinder. In this CPP instance we specified two required services for achieving the task. One is to calculate circular area and the other is to calculate multiplication of two values. *PatternProfile* is used to describe a CPP instance's functional and non-functional attributes as well as the types of the services which used to build it. A piece of code fragment of *PatternProfile* for describing the cylinder volume CPP is listed in figure 7. Note that the profile does not provide any inputs or outputs details for the required services. The reason for this is because the required services for building the pattern are only used to be invoked within the CPP instance. The details of required services are only described in *PatternModel* for automatic discovery purpose.

```
...
<rdf:Description rdf:resource="http://localhost/PatternExample.ows#CylinderVolumePattern">
  <rdf:type rdf:resource="http://localhost/ServiceOntology.ows#VolumePattern"/>
  <patternProfile:subService>
    <rdf:Bag> <rdf:Description rdf:resource="http://localhost/PatternExample.ows#CircularArea">
              <rdf:type rdf:resource="http://localhost/ServiceOntology.ows#AreaCalculation"/>
                </rdf:Description>
                <rdf:Description rdf:resource="http://localhost/PatternExample.ows#Multiplication">
              <rdf:type rdf:resource="http://localhost/ServiceOntology.ows#BasicCalculation"/>
  </rdf:Description>  </rdf:Bag>  </rdf:Description>
<patternProfile:hasInput rdf:resource="http://localhost/PatternExample.ows#PI"/>
<patternProfile:hasInput rdf:resource="http://localhost/PatternExample.ows#Radius"/>
<patternProfile:hasInput rdf:resource="http://localhost/PatternExample.ows#Height"/>
<patternProfile:hasOutput rdf:resource="http://localhost/PatternExample.ows#Volume"/>
...
```

Fig. 7. PatternProfile sample code.

PatternModel is an extended *ServiceProcessModel*. Compared with the original *ServiceProcessModel*, the *PatternModel* is specifically used to describe composition of services rather than processes. The process information is ignored here because the services are considered as whole abstract units to join the composition. The properties and syntax used to describe a composite process have been inherited from the *ServceProcessModel*. The code fragments in figure 8 illustrate how the *PatternModel* is used to describe the composition relationships between services. We can see that the mulitplicatio service in step2 uses the circular area service's output in the previous step as one of its inputs to calculate the final result for the cylinder volume.

ServiceGrounding only applies to concrete services rather than abstract services, so it is not part of CPP description.

```
...
<patternModel:Perform rdf:ID="Step2_Volume">
  <patternModel:subService rdf:resource="#Multiplication"/>
  <patternModel:hasdataFrom>
    <patternModel:InputBinding>
      <patternModel:toParam rdf:resource="#Multiplication_Input1"/>
      <patternModel:valueSource> <patternModel:ValueOf>
                <patternModel:theVar rdf:resource="#Area_Output"/>
                <patternModel:fromSubService rdf:resource="#Step1_Area"/>
      </patternModel:ValueOf>  </patternModel:valueSource>  </patternModel:InputBinding>
   ...
  </patternModel:hasdataFrom>
</patternModel:Perform> ...
```

Fig. 8. PatternModel sample code.

4 Implementation of the Common Process Pattern

To realize the CPP approach, a web based web service composition prototype has been developed. The main functions of this prototype include service browsing, build composite service, service flow execution, and generating extended OWL-S description for describing CPP instances. This prototype makes creating a CPP instance as easy as drawing a flowchart diagram. When the user try to connect two services together during the composition process, the prototype will give help information about whether these two services are compatible to be linked together. When a CPP instance is executed in the prototype, service matchmaking will be performed and all the required services are dynamically located based on the specifications. Due to the length limitation of the paper we have omitted the description of the matchmaking method, see [14] for details.

4.1 System architecture of the prototype

The system architecture of the prototype consists of four components: user interface (UI), service flow execution engine (SFEE), service searching engine (SSE), and semantic service pool (SSP), see figure 9.

The semantic service pool (SSP) component consists of three layers. The first layer is the WSDL described service pool. The services in the service pool could be from anywhere on the Internet. The second layer is the OWL-S description layer to add semantics into the WSDL service description. The third layer is the service ontology layer which organises the services in the service pool into a conceptual hierarchical structure.

The service search engine (SSE) component searches the OWL-S described service pool and locates the relevant services based on the requirements passed by the user interface or service flow execution engine. The SSE ranks the search results and returns it back to the user interface or the service flow execution engine. Normally the top ranking service is likely to be the service required by user or service flow execution engine, and the services with the lower ranking could be the alternative choice.

The service flow execution engine (SFEE) component executes the service flow created by the user and returns the executions result back to the user. The execution is based on the service flow control structure and the data dependency among services. Because the services within the service flow are abstract services which only have requirements specification, the service flow execution engine has to locate the concrete services through the SSE to execute the service flow.

The user interface (UI) component provides some utilities for a user to compose and run a service flow. It also provides an ontological display of service pool and a SSE interface for user to locate the required services.

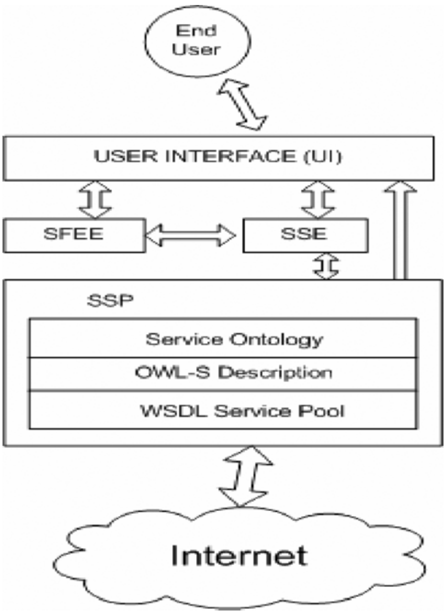


Fig. 9. System Architecture, arrows illustrate the relationship among components.

5 Related Work

Reuse is a major research area in software engineering field. Developers and designers realise how much effort and resource can be saved by enhancing the reuse technology. Therefore, many research works are focusing on reuse technology and pattern is one of the approaches. Gamma et al. [4] first identified the recurring elements in the object-oriented design and programming. Through three categories, they introduced 23 design patterns for design and constructing object-oriented program. Although these patterns are limited to use on object-oriented programming, the ideas of the pattern approach has been established.

Aalst et al. [1] worked on patterns for workflow construction. In order to address the complex workflow functionality, they provided 10 patterns and grouped them by four aspects, such as task synchronisation, common control structures, multiple instances of an activity, and state problems. All of these patterns are independent from the workflow language and workflow management system. These patterns focus on the low level workflow construction and execution and provide the common solutions for constructing workflow and monitoring workflow execution, but they cannot provide solutions for high level business process model.

Sirin et al. [12] presented a workflow template technique for dynamic web service composition. A workflow template is a generalised workflow and the activities within the workflow template are abstract. The idea of using abstract activity is to discover and substitute services at runtime. They use an extended OWL-S language to describe the abstract process and the matchmaking algorithm developed by Paolucci et al. [9] to match suitable service process with an abstract process. An extended HTN [13] planning formalism, HTN-DL, is used to describe how to generate the composition of web services. However, they did not address clearly how the workflow template can be created and used. Also the matchmaking algorithm makes it hard to locate a proper service.IBM [6] has provides several business patterns used to describe the relationship between the users, the business organizations or applications, and the data to be accessed. The patterns they provided are very high level patterns, such as self-service pattern (User to Business), information aggregation pattern (User to Data), collaboration pattern (User to User), and extended enterprise pattern (Business to Business). Those patterns provide solutions for the whole business rather than a common business process.

6 Conclusion and Future Works

Web service composition is important for reducing the developing time and cost of business services. An efficient and effective way to realize the composition is essential. In this paper, we proposed a CPP approach, to reduce the complexity of the process of service composition and requirement and task decomposition. A CPP instance contains abstract explicit specifications for each required service, so the concrete services can be located and invoked dynamically based on these specifications. Using bottom-up approach to build the pattern grantees that each of the CPP instances is executable.

Through the service composition research work, we realize that only integrate semantics into the service description cannot sufficiently address a service, the contextual information is also important for identifying services. In the future work, we will propose a contextual based semantic model to better describe services and a searching method based on this model to more accurately locate required services.

References

1. Aalst, W.M.P. van der, Hofstede, A.H.M. ter, Kiepuszewski, B., and Barros, A.P. (2003) Workflow Patterns, *Distributed and Parallel Databases*, Kluwer Academic Publishers, 14(1): pp. 5-51
2. Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S. (2001) Web Services Description Language (WSDL) 1.1,http://www.w3.org/TR/wsdl
3. Dustdar, S. and Schreiner, W. (2005) A Survey on Web Services Composition, Int. J. *Web and Grid Services*, Inderscience , Vol. 1, No. 1, pp.1–30.

4. Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995) Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Professional.
5. Ghallab, M., Nau, D., and Traverso, P. (2004) Automated Planning: Theory and Practice, Morgan Kaufmann, pp. 229-262.
6. Keen, M., Acharya, A., Bishop, S.n, Hopkins, A., Milinski, S., Nott, C., Robinson, R., Adams, J., and Verschueren, P. (2004) Patterns: Implementing an SOA Using an Enterprise Service Bus, IBM Redbook First Edition (July 2004), http://www.redbooks.ibm.com/abstracts/sg246346.html
7. Kumar, A., Srivastava, B., and Mittal, S. (2005) Information Modelling for End to End Composition of Semantic Web Services, in Proceedings of *4th International Semantic Web Conference (ISWC)*, 6-10 November 2005, Galway, Ireland.
8. Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., Mcllraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., and Sycara, K. (2004) OWL-S: Semantic Mark-up for Web Services, http://www.daml.org/services/owl-s/1.0/owl-s.html
9. Paolucci, M., Kawamura, T., Payne, T. R., and Sycara, K. (2002) Semantic Matching of Web Services Capabilities in Proceedings of the 1st *International Semantic Web Conference (ISWC)*, 9-12th June 2002, Sardinia, Italy.
10. Preist, C. (2004) A Conceptual Architecture for Semantic Web Services, In Proceedings of Third *International Semantic Web Conference*, Nov. 2004, Hiroshima, Japan, pp. 395-409.
11. Riehle, D. and Zullighoven H. (1996) Understanding and Using Patterns in Software Development, *Theory and Practice of Object Systems*, John Wiley & Sons, 2(1):3-13.
12. Sirin, E., Parsia, B., and Hendler, J. (2005) Template-based Composition of Semantic Web Services, 1st *International Symposium on Agents and the Semantic Web*, AAAI Fall Symposium Series Arlington, Virginia, USA 4th - 6th November, 2005.
13. Sirin, E., Parsia, B.,Wu, D., Hendler, J., and Nau, D. (2004) HTN planning for web service Composition using SHOP2, *Journal of Web Semantics*, Elsevier, 1(4), pp. 377–396.
14. W. Song (2006) A Semantic Modelling Approach to Automatic Services Analysis and Composition, to appear in the IASTED *International Conference on Web Technologies, Applications, and Services (WTAS)*, Calgary, Canada, 17-19 July.
15. W. Song (2006) Business Process and Integration Model: an Approach to Guide Constructing Service Flows, to appear in the IASTED International Conference on Web Technologies, Applications, and Services (WTAS), Calgary, Canada, 17-19 July.

Categories Extraction for Reuse in Semantic Applications and Profile Based Recommendation Service*

Vytautas Taujanskas *, Rimantas Butleris **

* Department of Information Systems, Kaunas University of Technology, Studentu St. 50-308, LT-51368 Kaunas, Lithuania
** Department of Informatics, Vilnius University Kaunas Faculty of Humanities, Naugardo g. 2, LT-44280, Kaunas, Lithuania

1 Semantic web – the new level of service

Consequent formalization of requirements for intelligent web came after the chaos with static, then afterwards with dynamic web resources. The semantic web is an extension of the current web, where information has well-defined meaning, better enabling computers and people to work in cooperation [**Hiba! A hivatkozási forrás nem található.**]. Abstract requirements [**Hiba! A hivatkozási forrás nem található.**] for information formalization became specific technologies [**Hiba! A hivatkozási forrás nem található.**,**Hiba! A hivatkozási forrás nem található.**] aimed to implement the vision of semantic web.

The rise of web 2.0 hype [**Hiba! A hivatkozási forrás nem található.**] is confirmed by the formation of critical mass of semantic applications. However, some of so-called web 2.0 applications take advantage of popularity and do not provide intelligent solutions. But the overall setting indicates the need of qualitatively new level of service, which could provide better functionality in *semantics*, *usability* and *collaboration*.

System must locate meaningful information resources on the Web and combine them in meaningful ways to perform tasks [**Hiba! A hivatkozási forrás nem található.**]. The process of locating or creating meaningful information should be

* The work is supported by Lithuanian State Science and Studies Foundation according to Eureka programme project “IT-Europe” (Reg. No 3473).

transparent to the end user. Various tools are needed to enrich data by integrating resources, dispersed across the web. The system itself must be intelligent to collect users' behavior patterns and to derive findings, which could be used to perform tasks.

This article is aimed to design web system according to the new generation of web application principles and to adapt the system for managing Lithuanian content. The study of internet activities of the Lithuanian users is performed in order to select the specific content for the system.

Collaboration functionality was taken into account, when designing the application. Collaboration is employed to build profile-based recommendations. The use of intelligent solutions, aggregation and other principles of the semantic web allowed us to reach the result by modeling service of recommendations.

2 Case study of local user

The idea of recommendation based system evolved in the context of research of semantic web technologies. The authors considered the idea of integrating personal data of Lithuanian web user. Case study of popular Lithuanian websites[1] was done to refine considerations about content.

Lithuania's online society has big variety of communication means: news portals (4 of 20 positions), lots of friendship services (5 of 20 positions) and places for communities, interested in some special knowledge domain (4 of 20 positions). The development of latter part of websites confirms the local formation of crowd wisdom [**Hiba! A hivatkozási forrás nem található.**] of some particular content categories (automobiles, family).

Widespread usage of friendship and communication services show popularity of communication systems. Though quality of functional characteristics of these systems in Lithuania, such as usability and intelligent way of presenting data, is quite low [**Hiba! A hivatkozási forrás nem található.**,**Hiba! A hivatkozási forrás nem található.**]. Most systems look like simple directories: they are defined in a rigid way, not discovered, are single-mode instead of multi-modal, connect users by means of shared artifacts [**Hiba! A hivatkozási forrás nem található.**]. Actually most of the systems do not have shared artifacts, just serve as directories for friends' contacts.

Summarizing main features Lithuanian internet user we can conclude, that he likes to read the news, shows trust to specialized communities and uses communication portals [**Hiba! A hivatkozási forrás nem található.**].

Therefore we intend to provide following functionality for our system:

- aggregate news content by using semantic technologies,
- implement user profiling functionality for extracting groups of likings (or wisdom) by using social networks,

[1] Lithuania's Top 20 2006 websites: http://www.ebiz.lt/article.php3/7/7993/0

- provide functionality of collaboration among the group members inside social networks.

3 Strategies for implementation

According to the functionality, as described above, the type of data of our system is defined as *Content*. As we choose data from news portal, the *Content* data element is the news article. Every *Content* instance is assigned to one or more *Categories* (Fig. 1). The *Categories* are used to define *User*.

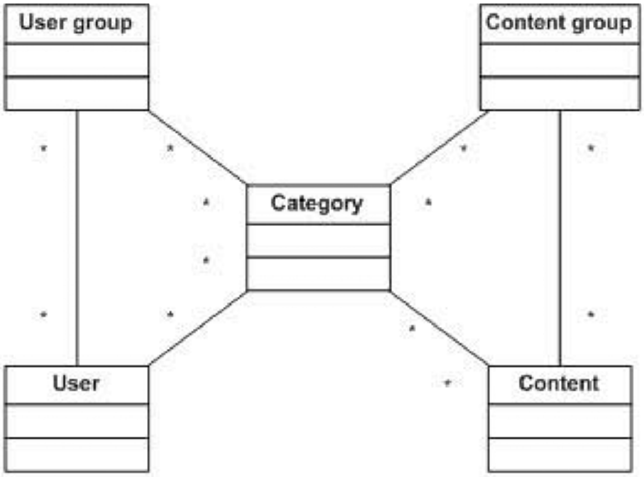


Fig. 1. Logic diagram for recommendation system

Associations among all entities in figure 1 have many-to-many multiplicity. Physical realization will use association class to reduce this complexity. However, *User* profile can be described by using multiple *Categories*, and *Category* can be used for multiple descriptions of *Users* likings. We use the same logic for describing *User groups'* likings, as the *Users* are assigned to *Users groups* according to related likings, described by *Categories*. Singular *User* can be assigned to multiple *User groups* according to likings match.

Analogical logic is used for defining *Content group*, *Content*, *Category relations*.

In this way, system can be divided into following modules:

- *Content* provider. This module is aimed to ensure content supply and provide functionality of simple categorization (as it is described in section 0 3.1 Providing the content). This part fills-in the *Content* and *Category* entities.

- Categorization module. This module provides more intelligent and accurate categorization functionality by filling *Category* entity. This process is described in section 0 3.2 Categories for user profiles building.
- *User* profiler. This module defines user profile according to actions of user in the system. For example, most preferred categories are used to describe user likings. In the next stage of using the system other types of information can be aggregated, such as extracted user data from popular portals, mentioned in section 0 2 Case study of local user.
- Recommendation service. This module analyzes data provided in the *User* profile description, links it to the other users of *User group*. Service has to provide related Content according to *Categories* profile description. *Content Categories* are used to define *Content groups* and provide dynamic flow of *Content*. The strategies for recommendations are described in section 0 3.3 Recommendations as a part of collaborations.

3.1 Providing the content

The main problem of building recommendation system is content aggregation. We chose to redirect the received content, which is available through RSS (Rich Site Summary/ Really Simple Syndication) or other format feeds [**Hiba! A hivatkozási forrás nem található.**]. The most popular Lithuanian news portals provide separate feeds for different categories (daily, business, sports, entertainment, life-style and subcategories). That helps us to define initial categorization as presented in figure 2. During initial launch of system we chose the strategy of providing simple categories management, by using easily extracted, though maybe not so accurate categories.

More complex categorization can be done by analyzing data provided in the feed. For example, one of content providers displays categories name in brackets before title *element* in each *item*. For getting better results the authors of [**Hiba! A hivatkozási forrás nem található.**] analyze various types of sources: meta-data, URL.

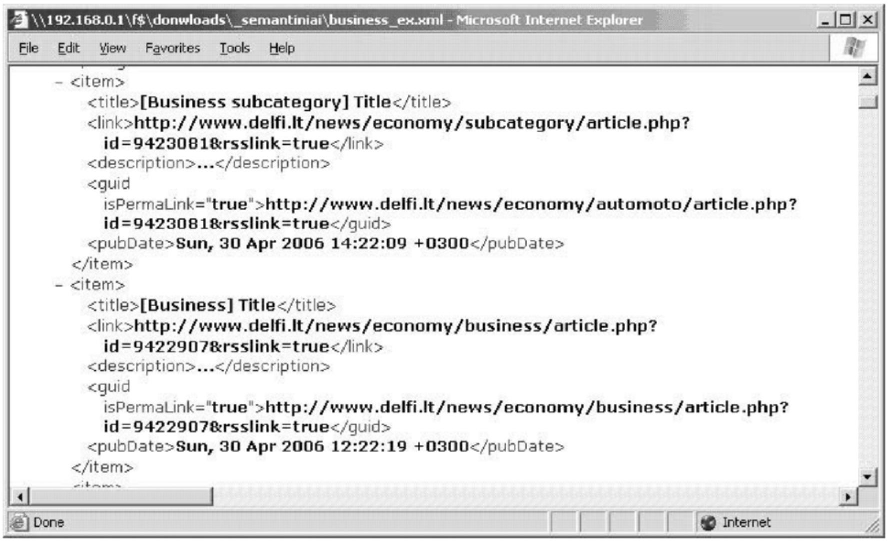


Fig. 2. News feed example

Other necessary data, which is not available in feeds, can be parsed using XPath queries for DOM[2] (Document Object Model). At the time of writing the article only one of 4 most popular Lithuanian news portals (section 0 2 Case study of local user) did not provide feeding service. XPath queries can be used to supplement feed with link, name and category of the content.

For successful functioning of the system, the need of tracking content structure changes is indicated. This functionality is planned for later implementations and should include auto disabling of erroneous scripts for building feed.

3.2 Categories for user profiles building

However, in order to provide better service of categorization and, therefore, better description of user likings, the system needs more intelligent category building solutions. For this purpose the reusable solutions[3] for analysis of English texts can be used. For analysis of Lithuanian texts the ready-to-use solutions do not exist yet. Lithuanian language is inflective and therefore it implies specific morphological and ambiguity removing issues to be solved.

Category building function can be assigned to system users themselves [**Hiba! A hivatkozási forrás nem található.**], but this approach incurs risk of lack of user input for qualitative categorization, caused by quickly changing content.

[2] http://www.w3c.org/DOM

[3] http://www.topicalizer.com

One more available method is automatic categorization [**Hiba! A hivatkozási forrás nem található.**] based on machine learning with preset[4] or emerging categories.

The lack of functionality for Lithuanian language, similar to WordNet [**Hiba! A hivatkozási forrás nem található.**], limits application of semantic relation queries to this language. Some useful relations shown in table 1 could be used for analyzing categories: generalization, association building and others.

Table 1. Semantic relations available in WordNet

Relation	Definition	Example
Hypernym	from concepts to subordinate;	breakfast -> meal
Has-Member	from groups to their members;	faculty -> professor
Member-Of	from members to their groups;	copilot -> crew
Has-Part	from wholes to parts;	table -> leg
Part-Of	from parts to wholes;	course -> meal
Antonym	opposites;	leader -> follower
Homonym	same form, but different meaning	bank: river bank, financial institution
Polysemy	same form, related meaning	bank: blood bank, financial institution)
Synonymy:	different form, same meaning	singer, vocalist

However there are some solutions, available for morphological [**Hiba! A hivatkozási forrás nem található.**] and ambiguities removing [**Hiba! A hivatkozási forrás nem található.**] features of Lithuanian language. The motivation to put these solutions on the web guides us to use machine-based categorization or simple frequent mean words list describing the article.

Implementation of publications categorization and multiple user aware system could allow us to examine available information and employ it for collaborative filtering.

3.3 Recommendations as a part of collaborations

Collaborative filtering is the method, used to predict interests of a user by collecting similar information from other users. The main assumption of this approach states, that the group of users which used to like the same thing will like the same thing in the future. In the [**Hiba! A hivatkozási forrás nem található.**] we use this approach to define user groups in social networks.

Collaboration, as a part of social network installed in the system, is provided by commercial and non-commercial systems AmphetaRate, LastFm and etc[5].

[4] http://www.dmoz.org/World/Lietuvių/

[5] www.amazon.com, www.last.fm, www.stumbleupon.com and http://amphetarate.newsfairy.com/

AmphetaRate focuses on the items found on dynamic feeds. It uses Bayesian statistics [**Hiba! A hivatkozási forrás nem található.**] method for choosing the items which could possibly be liked by the user. Generally, the Bayesian approach is used for spam filters, and AmphetaRate uses it for ratings. Another RSS rating approach, used in AmphetaRate and also in other collaborative systems, is called collaborative filtering [**Hiba! A hivatkozási forrás nem található.**] (often abbreviated as ACF). Functionality installed in AmphetaRate includes:

- Person related ratings. Rating estimation depends on ratings of the user.
- Employs the social factor to define relations. as it uses people to determine relations.
- Provides fortunate discoveries by accident.

Authors of [**Hiba! A hivatkozási forrás nem található.**] provide comprehensive research of recommendation systems. They separate three classes of such systems:

- Content-based recommendations, where the user is recommended according to the items which he preferred in the past.
- Collaborative recommendations, where the user is recommended according to the items, which were liked by people with similar tastes and preferences in the past.
- Hybrid approaches, which combine both of above.

The research enumerates possible extensions of recommender systems capabilities, which could take into account *more comprehensive understanding of the user*, analysis of user's transactional, navigational web usage patterns and other available data.

The system should be non-intrusive, as it could measure the amount of time, spent by the user for reading an item, and also should be accurate for the noise. One way of exploring the *non-intrusiveness* is determining the optimal number of ratings. Other way is to develop marginal cost model (instead of the fixed cost) and to provide the cost/benefit analysis. Authors [**Hiba! A hivatkozási forrás nem található.**] propose active learning methods to address this issue.

Interesting flexibility approach is proposed, where RQL (Recommendation Query Language) is used to resolve the flexibility issue. Despite of the method, used to solve it, it should *provide recommendation not for the particular items, but for brands or categories of items to the corresponding segments of users*.

4 Conclusions

The system, which aggregates information from the web and analyzes it by adding recommendations functionality, is proposed. In order to select sources for aggregation the local internet user analysis is made. Categorization functionality is added for description of content and user profile. Initial categories extraction is implemented by multi-way analysis of incoming data feeds. More intelligent solutions adapted to Lithuanian language are substantiated for later use.

Recommendation system contains available categories and uses them to describe user's profile. Automatic extraction of categories has advantage over tagging, as it can ensure non-intrusiveness, which is essential aspect.
More extensive user oriented analysis should be done by providing other data aggregation (i.e. friendship directories) or transactional actions recording. Rating is liable for correct recommendations as well.

References

1. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, vol 17, issue 6, ISSN: 1041-4347, pp 734 – 749
2. Bayesian statistics. http://en.wikipedia.org/wiki/Bayesian_statistics
3. Berners-Lee T, Hendler J, Lassila O (2001) The Semantic Web. Scientific American. http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2
4. Boyd DM (2004) Friendster and publicly articulated social networking. In CHI '04 Extended Abstracts on Human Factors in Computing Systems, ACM Press, New York, NY, pp 1279-1282
5. Decker S, Melnik S, van Harmelen F, Fensel D, Klein M, Broekstra J, Erdmann M, Horrocks I (2000) The Semantic Web: the roles of XML and RDF. Internet Computing, IEEE, vol 4, no 5, ISSN: 1089-7801, pp 63–73
6. Fisher D (2003) Social Networks for End Users. Survey Paper for Advancement to Candidacy. University of California, Irvine. http://www.bsos.umd.edu/gvpt/CITE-IT/Documents/Fisher 2003 Soc Ntwks for End Users.pdf
7. Gupta KM, Aha DW, Marsh E, Maney T (2002) An architecture for engineering sublanguage WordNets. In Proceedings of the First International Conference On Global WordNet, pp 207-215
8. Mika P (2005) Flink: Semantic Web Technology for the Extraction and Analysis of Social Networks. Journal of Web Semantics, vol 3, no 2, pp 211–223
9. Ohmukai I, Hamasaki M, Takeda H (2005) A Proposal of Community-based Folksonomy with RDF Metadata. In Proceedings of the ISWC 2005 Workshop on End User Semantic Web Interaction (2005), published on CEUR-WS
10. O'Reilly T (2005) What is Web 2.0. http://www.oreilly.com/go/web2

11. Rimkute E, Grybinaite A (2004) The most frequent types of the morphological ambiguity of the Lithuanian language and the automatical disambiguation of them (in Lithuanian). Kalbu studijos, vol 5, pp 74 – 78
12. Sebastiani F (2002) Text categorization in general - a survey paper: Machine Learning in Automated Text Categorization. ACM Computing Surveys, vol 34, no 1, pp 1 – 47
13. Surowiecki J (2005) The Wisdom of Crowds. Anchor, ISBN:0385721706
14. Svatek V, Berka P, Kavalec M, Kosek J, Vavra V (2003) Discovering Company Descriptions on the Web by Multiway Analysis. Springer-Verlag, Advances in Soft Computing series, ISBN 3-540-00843-8, pp 111 – 118
15. Tepper M (2003) The rise of social software. netWorker, vol 7, no 3, ISSN:1091-3556, pp 18–23
16. Tutkutė L, Taujanskas V (2006) Forming recommendations in semantic web (in Lithuanian). Information technologies for business and study 2006. vol 1, pp 178–183
17. Ungar L, Foster D (1998) Clustering Methods for Collaborative Filtering. In Proceedings of the AAAI-98 Workshop on Recommender Systems, AAAI Press, pp 112 – 125
18. Uschold M (2003) Where are the semantics in the semantic web? AI Magazine, vol 24, issue 3, ISSN:0738-4602, pp 25 – 36
19. Wusteman J (2004) RSS: the latest feed. Library Hi Tech, vol 22, no 4, MCB University Press, ISSN 0737-8831, pp 404 – 413
20. W3C. Semantic web activity. http://www.w3.org/2001/sw/
21. W3C. SPARQL Query Language for RDF. http://www.w3.org/TR/rdf-sparql-query/
22. Zinkevicius V (2000) Morphological analysis with Lemuoklis (in Lithuanian). Darbai ir Dienos, vol 24, pp 245 – 273

Knowledge and Decision-Making within Software Projects

*Birinder Sandhawalia, **Darren Dalcher

*School of Computing Science, Middlesex University, Trent Park, Bramley Road, London N14 4YZ, UK. b.sandhawalia@mdx.ac.uk
**School of Computing Science, Middlesex University, Trent Park, Bramley Road, London N14 4YZ, UK. d.dalcher@mdx.ac.uk

Abstract: The effective use of knowledge results in better decision-making within projects. Moreover, the unpredictable nature of software projects and the need for effective communication within project teams requires a framework for social interaction and feedback that results in better decision-making. This paper analyses the creation and capture of knowledge within software development projects and discusses the central role of decision making in the development process, and how the effective use of knowledge helps improve decision-making. The paper views how the knowledge generated and decisions made within a software project can be provided greater visibility and communicated effectively, and to achieve this, presents a framework to facilitate social interaction and feedback during the development process.

1 Introduction

The use of knowledge is expected to result in better decision-making, innovation and competitive advantage within organisations and projects. While organisations are permanent structures that emphasise functional structure, projects are temporary endeavours that are predicated on a deadline and the ultimate delivery of an artifact, product, service or quantified business value or benefit. As part of their normal operations, organisations capture knowledge in repeatable situations that are relatively straight-forward and make-sense. Projects, by definition, are focused on change. They function in an uncertain and ambiguous environment, and need to be flexible and adaptive. The temporary nature of projects and their deadline driven schedules require a decision-making process that lies at the core of the project processes to ensure that they are executed smoothly. Projects need reflection, practitioner's experience, and the ability to build on acquired knowledge.

Software development projects are life-cycle driven and are organised around teams that are assembled specifically for the limited duration of the project. The software development process relies on the knowledge and creativity of individuals and teams, and the formation of these teams requires the involvement and participation of all team members in the development process. There is also an increasing need to involve users early in the software development life-cycle since designing software requires extracting detailed knowledge of the users. Effective communication is the basis for discussion between users and developers during the requirements definition process that is essential to provide an understanding of the software requirements. However, problems of communication occur due to the diversity of professional expertise and organisational roles that confer users' different views and expectations of the system to be developed.

The unpredictable nature of software projects and the need for effective communication within project teams necessitates a framework for social interaction and feedback that results in better decision-making. This paper analyses the creation and capture of knowledge within software development projects. The paper discusses the central role of decision making in the development process and how the effective use of knowledge helps to improve decision-making during the development process. The knowledge created and decisions implemented need to be effectively communicated across the entire process. Social interaction and feedback are key factors that facilitate the effective use of knowledge within software projects. The paper views how the knowledge and decisions implemented can be provided greater visibility within the projects and communicated effectively, and also presents a framework to facilitate social interaction and feedback during the development process.

2 Knowledge

Knowledge is the capacity for effective action. Alavi and Leidner (1999) define knowledge as 'a justified personal belief that increases an individual's capacity to take effective action.' While 'personal' implies the contextual nature of knowledge, action requires competencies and know-how, and implies the dynamic nature of knowledge. Knowledge is fluid and formally structured, and it exists within people, processes, structures and routines, (Davenport and Prusak 1998). Polanyi (1967) suggests that knowledge exists as tacit and explicit. Tacit knowledge comprises an individual's mental models, and while it is personal and in the mind of an individual, it is also context specific and difficult to articulate, formalise and verbalise, and is therefore hard to communicate and share. The factors that influence an individual's mental model include the individual's education, expertise, past experiences, perceptions, biases, prejudices and environment. Explicit knowledge can be easily articulated and codified and therefore transmitted and communicated. Polanyi (1967) contends that human beings acquire knowledge by actively creating and organising their own experiences and sums it up by stating that "we can know more than we can tell."

The importance of knowledge is increasing as organisations recognise that they posses knowledge and increasingly learn to view this knowledge as a valuable and strategic asset. Knowledge assets include knowledge which resides within the individuals, systems, processes, documents and structures of the organisation. Davenport and Prusak (1998) recommend that to remain competitive, organisations must efficiently and effectively create, capture, locate and share their organisations knowledge and expertise, and have the ability to bring that knowledge to bear on problems and opportunities.

2.1 Knowledge Management

The American Productivity and Quality Center (1996) defines knowledge management as "a conscious strategy of getting the right knowledge to the right people at the right time and helping people share and put information into action in ways that strive to improve organisational performance." Knowledge management, therefore, requires that it is imperative to identify what knowledge needs to be managed, how, when, where, by whom, and for whom. Consequently, the key elements of KM are collecting and organising the knowledge, making it available through knowledge infrastructure, and then using the knowledge to improve decision making and gain competitive advantage. Alavi and Leidner (1999) refer to knowledge management as a systematic and organisationally specified process for acquiring, organising and communicating both tacit and explicit knowledge of employees so that other employees may make use of it to be more effective and productive in their work and decision-making while improving product and process innovation.

3 Decision Making

Decision making is a crucial part of organisational as well as personal activity. Simon (1977) states that decision making comprises four principal phases which are: finding occasions for making a decision, finding possible courses of action, choosing among courses of action, and evaluating past choices. Simon suggests that the four phases be called intelligence, design, choice and review activities. The intelligence activity involves searching the environment for conditions calling for decision, while the design activity comprises inventing, developing and analysing possible courses of action. The choice activity entails selecting a particular course of action from those available, and the review activity assesses the choices made. The four phases are similar to those suggested by Nutt (1989) who states that the decision making process requires exploring possibilities, assessing options, testing assumptions and learning. As KM attempts to make available 'the right knowledge to the right person at the right time', it improves the processes of exploring possibilities, assessing options and testing assumptions.

Simon (1977) further distinguishes between programmed and non-programmed decisions. Well-structured, repetitive and routine decisions are programmed so that a definite procedure for handling them exists, while non-programmed decisions are novel, unstructured and unusually consequential. Programmed decisions are addressed by routine standard operating procedures, while non-programmed decisions require human judgment, insight and intuition. Therefore while programmed decisions would benefit from the explicit knowledge of an organisation, non-programmed decisions require creativity and rely upon the tacit knowledge of the individuals.

Knowledge provides the context and insights required to improve the quality of information used in the decision making process, as knowledge is considered to be authenticated information viewed with context. Knowledge improves the ability to respond to the sub-merged problems and issues and helps in identifying and evaluating several courses of action before committing to one while exploring possibilities. Knowledge enhances the ability to identify factors that influence estimates used to value the merits of alternative courses of action while assessing options. Knowledge is also useful to examine the value attached to each option or alternative while testing assumptions. Reflecting upon the outcome of the decisions made and identifying missed opportunities results in learning and provides important insights that lead to the creation of further knowledge. This dynamic creation of knowledge, where knowledge created is the basis for further knowledge creation, is made possible through feedback and interaction of knowledge (Nonaka and Takeuchi (1995). The knowledge created provides the basis to explore further possibilities, set objectives, and identify ways to respond thereby resulting in more effective decision-making within organisations and projects.

4 Projects

Compared to organisations which are permanent structures and have routines, projects are temporary by nature and their implementation requires creative actions, practitioner's experience, and the ability to apply knowledge to development problems. Projects are designed to achieve specific objectives within a predetermined time frame, budget and resources. Projects involve planning for non-routine tasks to be conducted in several phases, and can be characterised as unique, goal-oriented and complex undertakings steeped in uncertainty, which aim to produce a meaningful product or service in order to satisfy a need, (Dalcher 2003). Davis (1951) defines a project as "any undertaking that has definite final objectives representing specified values to be used in the satisfaction of some need or desire." The Project Management Institute (PMI) makes a distinction between ordinary work and projects by emphasising the temporary and unique nature of projects (PMBOK 2000).

Bredillet (2004) states that for the past forty years projects and their management have become a well-accepted and strategic way to manage organisations. According to him, projects as strategic processes modify the conditions of the organisation and its environment. Meredith and Mantel (1995) describe three basic types of project organisations:

- projects as a functional section of a larger organisation
- pure project organisations where a project is accommodated in a separate, largely self-contained section that is disbanded when the project is completed or disbanded
- organisations where projects are run on matrix basis in which control rests with a project manager but the majority of human and other resources are borrowed from different sections of the larger organisation

Knowledge is required to implement the project in relation to the type of project organisation. Koskinen (2004) suggests a metaphor of a project tree to visualise the entire knowledge required by a project organisation and states that the types of knowledge that a project may require are tacit, explicit, additive or substitutive. Koskinen refers to additive and substitutive knowledge as knowledge that is new to the project and is either invented internally or acquired from external sources. This is similar to Bredillet's (2004) view that project teams need to know what knowledge is available to complete the project based on past experience, and what knowledge needs to be acquired or will emerge as a result of the unique nature of the project tasks, especially within software projects.

4.1 Software Projects

Software projects are life cycle driven and follow the sequence of going from concept through definition and development, to testing and delivery. However, unlike other projects, the requirements of software projects are subject to frequent change. As a product, software can be changed, and it is therefore assumed that this change is possible at even the later stages of the development process. Such change and uncertainty make software projects more unpredictable than other projects, and are therefore organised around teams, relying upon the knowledge and creativity of the individuals and the teams.

Myers (1985) states that more than half the cost of complex software development is attributable to decisions made in the 'upstream' portion of the development process, namely, requirements specification and design. Traditional software development models like waterfall and spiral follow a sequential approach where the 'upstream' and 'downstream' aspects of development are clearly distinguishable. However, modern approaches like agile methods are more emergent and evolutionary, and rely on frequent feedback and interaction between and within self-organised teams while attempting to address change and uncertainty in the requirements. Self-organised teams comprise a group of peers who posses a sense of ownership and share responsibility for managing their own work, which includes problem-solving and continuous improvement of work processes. Gruenfeld et al (1996) and Politis (2003) are both of the view that self-organised teams create and share new knowledge, resulting in improved decision making and, furthering the ability of the teams to deliver the best possible solutions. Knowledge is the raw material required for decision making within software design teams and for complex projects, knowledge from multiple technical and functional domains is required (Curtis et al 1988). Walz et al (1993) state that ideally, software development teams are staffed so that both the levels and distribution of knowledge within them match those required for the successful completion of the project. Software development processes require that the individual activities are co-ordinated towards the team activities, focusing on the different areas and stages of the developmental process. The software development process involves eliciting the user requirements and software satisfying the requirements is then designed, built, tested and delivered.

Process models for software development depict sequential, incremental, prototyping or evolutionary approaches. Developmental models help simplify and reduce the complexity within software projects by providing a perspective to organise the different stages or phases of the development process. However, as stated by Myers (1985), decisions made early in the software development process have an effect in the later stages of the process. Software projects involve dealing with trade-offs between characteristics, preferences and quantities, while maintaining a balance between requirements, expectations, perceptions, opportunities and risks. Therefore software projects require a framework that enables the use of knowledge to facilitate decision making within the process to achieve the desired outcome. The next section presents a model that provides the framework for decision making within the developmental process.

5 The Dynamic Feedback Model

Complex and uncertain software development situations require a model that can account for the knowledge needed to plan and implement decisions within the development process. An example of such a model is the Dynamic Feedback Model (DFM) that underlines the relationships and interactions between the entities by depicting the feedback loops operating between them. The model, as depicted in Figure 1, focuses on four different functional areas that are intertwined throughout software development. The DFM models the relationships in a non-linear fashion amongst the functional areas and allows a continuous view of the development process. The four areas are management, technical, quality and decision making.

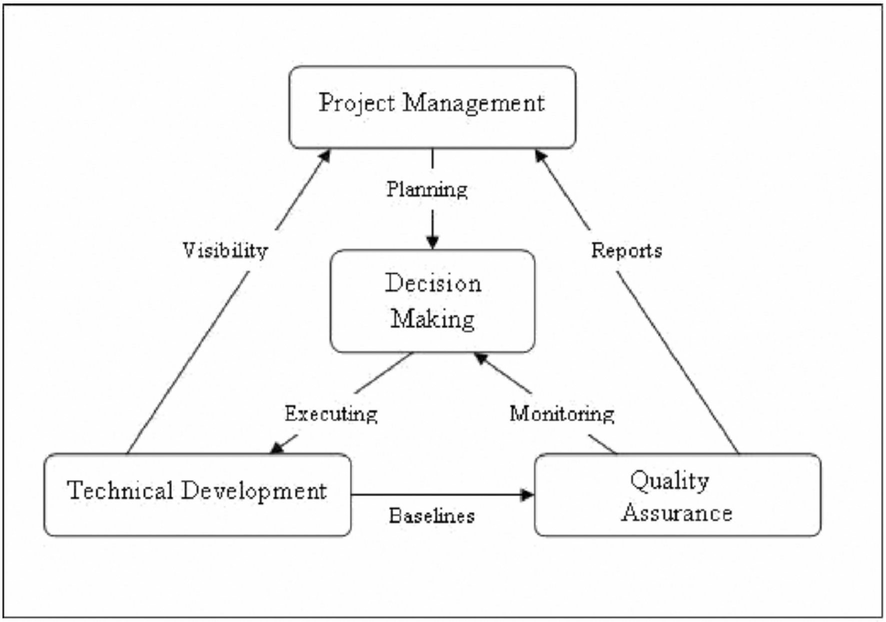


Fig. 1. Dynamic Feedback Model

5.1 Functional Areas

The management area involves the planning, control and management of the software development process. It also pertains to the strategy and operation of the project. Key concerns revolve around identifying performance gaps, assessing progress, and allocating resources to accomplish tasks. As technical development and knowledge creation are on going activities, the management area also takes on a continuous view. It does not limit its focus to delivering a product, but to the continuous need for generating and maintaining an on-going flow of knowledge required for continuous development.

The technical area deals with the creation and continuous improvement of the software system. The area recognises the changing needs and perceptions of the development process. The activity in this area includes evolution and maintenance of the software, while also maintaining its functionality and utility. Experimentation, learning and discovery take place as the software goes from inception to evolution. The development and design of the software form the basis for interaction between team members, and the knowledge created through the interaction provides the raw material for decision making within the process.

The quality area is perceived as a dynamic dimension, which continuously responds to perceived mismatches and opportunities reflected in the environment. It is concerned with assuring the quality of the product developed and the process used to develop it. Being an area of assessment, it provides the basis for learning.

The decision making area lies at the core of the model as software development is described as a decision making process (Dym and Little 2000). This area attempts to balance knowledge, uncertainty and ambiguity with a view to maximise the expected returns on an on-going basis. Knowledge acquired from implementing decisions is used within the process either as background knowledge available to support future decisions, or as a formalised part of an integral body of knowledge which can be used to optimise the decision making process. Decision making helps manage opportunity and risk and therefore this area can also be considered the risk management area. Risk assessment and planning are key activities within this area, which also ensures the implementation of decisions and the monitoring of their execution on a continuous basis. The knowledge and information required for the implementation, execution and monitoring of decisions is provided by the interaction and feedback loops of the model.

5.2 Feedback Loops

The DFM is in essence a set of interactions and feedback loops governing and controlling the development of software form a continuous perspective. The decision making perspective of the DFM ensures that rational and reasoned choices are made from the alternatives available during the development process.

The basic loop in the dynamic system is the planning-control-visibility loop. This loop helps to plan and control the production, evolution and growth of the software in association with project management and decision making. The loop enables the continuous generation of new information as well as feedback knowledge. The use of this knowledge is crucial in changing plans to adapt to reality and opportunities, modifying the decisions and re-examining the assumptions. The visibility of this basic feedback loop provides a continuous process to ensure the system remains relevant with regard to its objectives.

The configuration control loop links control, baseline and monitoring that provide the link between monitoring and controlling. Monitoring provides feedback on the strategies implemented, and is the link between decision making and quality. The monitoring loop is a closed–loop system providing feedback to the decision making area and ensuring effective control of the decisions made in technical development while relying upon quality assurance techniques and feedback mechanisms to evaluate progress and quality.

The reporting planning loop provides visibility to the project management area regarding the opportunities and mismatches present in the quality area and also provided by the implementation and execution of the decisions made.

The above mentioned loops depict relationships between the different functional areas. The DFM can therefore be used as a framework for understanding the dynamic nature of the interactions between entities in software development. The model moves away from linear thinking and offers a continuous perspective for understanding and implementing relationships, and the role these relationships play in decision making. The model achieves this through the on-going feedback and interactions of the loops, which present the framework to provide knowledge required for decision making. The following section examines the feedback and interactions between the different phases of software development.

5.3 The DFM Process

The phases of the software development process can broadly be categorised as problem definition, requirements analysis, design, implementation and maintenance. The DFM views knowledge as a key asset in the development of software and focuses on its feedback within the functional areas of development. In doing so, the DFM encourages thinking about software development in terms of the different phases and their interactions. The feedbacks within the functional areas of the DFM are depicted in Figure 2, and the use of knowledge for decision making within the various activities of software projects are discussed there-after.

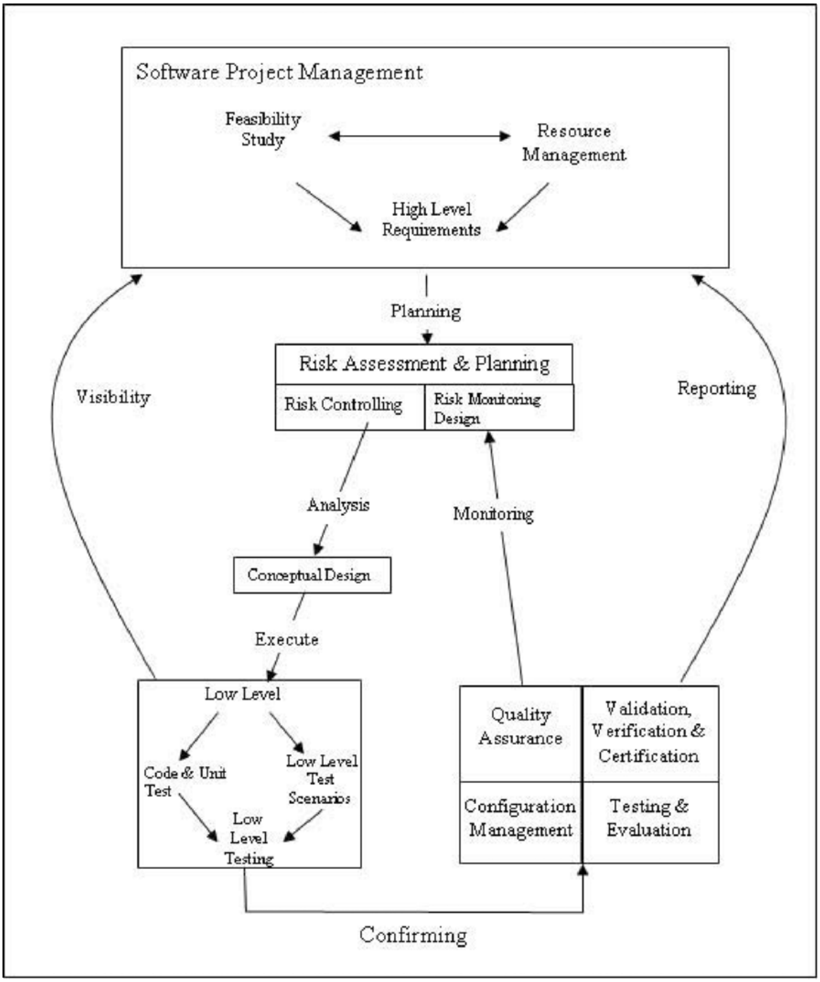


Fig. 2. Feedback of DFM

The project management area facilitates project planning and execution, and is also where the user requirements are elicited and the problem defined. Planning involves resource management where the skills and competencies required to execute the project are identified and teams are formed. Proper requirements analysis and specification are critical for the success of the project, as most defects found during testing originate in requirements. In order to understand the customer requirements, the developers require insight into the domain of the business system and the technical concepts of the system to be developed. Knowledge is created while understanding the requirements by the interaction of the different team members, and also between the users and the developers. This knowledge provides the perspective for decisions made to implement the project. The project management area is where negotiation takes place between the users and developers of the software. Johnson (1993) defines negotiation as a process in which individuals or groups seek to reach goals by making agreements with each other. Software development requires that users are involved in the development of the software. A clear understanding is needed between the users and developers to build the software, and this understanding is established through dialogue and negotiation. The formalisation of such an understanding usually results in the form of proposals and contracts. The feasibility of the project and the cost involved in executing the project are the basis for the proposals and contracts. The project management area addresses the need to assess the feasibility of the project and its cost analysis.

Based upon the decisions made and the outcome of planning within the project management area, an analysis of the impact the project will have on the business and technical environment is made along with the possible risks involved in implementing the project. The analysis views the goals, scope and functionality of the system being developed and how they fit or respond to the existing processes with which they are required to interact. Risk assessment and planning are conducted and feature the two traditional components of risk identification and prioritisation. Identification tries to envision all situations that might have a negative impact on the project, while prioritisation involves analysing the possible effects and consequences of the risk in case it actually occurs. The project also requires crucial decisions to be made in the design stage. High level design is the phase of the life cycle that provides a logical view of the development of the user requirements. Design involves a high level of abstraction of the solution, through which requirements are translated into a 'blueprint' for constructing the software, and provides the architecture of the application and its database design. Decision making at this stage of the process helps transform the requirements into a set of software functions and a physical database structure. Scenarios are developed to test the acceptability of the design with relation to the requirements.

The technical activities of design, code generation and testing are performed in the technical area. The area includes the detailed design phase where the high level design is converted into modules and programs. A unit test plan is created for the conditions for which each program is to be tested. The required programs are coded or translated into the programming language, and the programs are tested using the unit test plans. The technical area ensures that the integration plan is implemented according to the environments identified for integration. The area also ensures the maintenance, functionality and utility of the software apart from its creation and evolution. The decisions made in this area relate to the technical activities and provide confirmation of the design and suitability of the requirements. The decisions made are verified during system testing within the quality assurance area.

Pressman (1997) states that quality assurance consists of the auditing and reporting functions of management, and that its goal is to provide management with the data necessary to be informed and assured about product quality. The quality assurance area involves system testing which validates that the software developed meets the requirement specification. This phase identifies the defects that are exposed by testing the entire system. A series of tests are performed together, each with a different purpose, to verify that the system has been properly integrated and performs its functionality and satisfies the requirements. The quality assurance area thus provides verification of the decisions made and tasks performed in the technical area while confirming the decisions made during the design phase, and validating the requirements.

The different phases of the process are validated and given visibility by the feedback loops. Controlling the execution of decisions generates knowledge (Dalcher 2003a). The feedback loops globalise this knowledge within the process and ensure that knowledge is available for decision making. The decisions made in the decision making area during design and risk analysis receive confirmation during technical development and quality assurance. Technical development provides the project management area visibility of the software being developed to meet the requirements. Quality assurance further reports and validates to project management the decisions made during design and technical development. The project management area is able to assess the feedback and incorporate it in planning to help address some of the change and uncertainty inherent within the software development process

6 Conclusions

The DFM model provides a decision-making perspective and the execution of decisions helps create new knowledge. The knowledge created further helps support decision making which lies at the core of the DFM process. Knowledge is globalised to the functional areas of the process through feedback loops. The loops represent the knowledge intensive activities which connect decision making to other phases of development process.

The DFM adopts a long-term perspective of software development that enables it to address the issues of uncertainty and ambiguity, and therefore benefit from the decisions made and knowledge created during the development process. The long-term perspective also enables the DFM to look beyond a single project and use the knowledge generated towards improvements in future software projects. The DFM is receptive to changes in the environment and tackles them by feeding acquired knowledge back into the decision making process. As software development becomes more integrated in management practices the importance of continuous learning, knowledge, and skill acquisition as underpinned by the DFM will remain central to improved control, visibility and management. The availability of a long-term view justifies the adoption of multiple perspectives, the reuse of knowledge and the utilisation of a decision making perspective, which underpin feedback and improvement. The DFM provides a framework that facilitates social interaction and feedback, which further enhance the use of knowledge for decision making within the software development process.

References

1. Alavi M and Leidner DE (1999) 'Knowledge Management Systems: Issues, Challenges and Benefits,' Communications of the Association for Information Systems, Vol 1, Article 7.
2. American Productivity and Quality Center (APQC) (1996), Knowledge Management: Consortium Benchmarking Study Final Report; Available from http://www.store.apqc.org/reports/Summary/know-mng.pdf
3. Bredillet CN (2004) 'Projects are Producing the Knowledge which are Producing the Projects……' Proceedings of IPMA, Budapest.
4. Curtis B, Krasner H, and Iscoe N (1988) 'A Field Study for the Software Design Process for Large Systems,' Communications of the ACM., Vol 31, No 11, pp 68-87.
5. Dalcher D (2003) Computing Project Management, Middlesex University Press, London.
6. Dalcher D (2003a) 'Software Development for Dynamic Systems.' Developments in Metainformatics. LNCS, Springer Verlag, Peter J Nurnberg (ed): pp 58-75.
7. Davenport TH and Prusak L (1998) 'Working Knowledge,' Harvard Business School Press, Boston.
8. Davis RC (1999) 'The Fundamentals of Top Management,' New York: Harper and Brothers, 1951, (quoted in Cleland DI. Project Management: Strategic Design and Implementation, 3rd ed. New York: McGraw-Hill, p-3.
9. Dym C L and Little P (2000) 'Engineering Design: A Project Based Introduction.' John Wiley, New York.
10. Gruenfeld DH, Mannix EA, Williams KY, and Neale MA, (1996) 'Group Composition and Decision Making: How Member Familiarity and Information Distribution Affect Process and Performance,' Organisational Behaviour and Human Decision Process, Vol 67, pp 1-15.
11. Johnson RA (1993) 'Negotiation Basics,' Sage Publications, Inc.
12. Koskinen KU (2004) 'Knowledge Management to Improve Project Communication and Implementation,' Project Management Journal, Vol 35, No 1, pp 13-19.
13. Meredith JR and Mantel JS (1995) 'Project Management: A Managerial Approach,' Wiley, New York.
14. Myers W (1985) 'MCC: Planning the Revolution in Software,' IEEE Software.
15. Nonaka I and Takeuchi H (1995) 'The Knowledge Creating Company: How Japanese Companies Create the Dynamics of Innovation,' Oxford University Press, New York.
16. Nutt PC (1989) 'Making Tough Decisions,' Jossey-Bass Inc.
17. Polanyi M (1967) 'The Tacit Dimension,' Routledge and Keon Paul, London.

18. Politis JD (2003) ‘The Connection between Trust and Knowledge Management: What are its Implications for Team Performance,’ Journal of Knowledge Management, Vol 7, No 5, pp 55-66.
19. Pressman RS (1997) ‘Software Engineering – A Practitioner’s Approach,’ The McGraw-Hill Companies, Inc.
20. Simon H A (1977) ‘The New Science of Management Decision,’ Prentice-Hall, Inc.
21. Walz DB, Elam JJ and Curtis B (1993) ‘Inside a Software Design Team: Knowledge Acquisition, Sharing and Integration,’ Communications of the ACM, Vol 36, No 10, pp 63-77.

Index

D

T

U

V