

Weiming Shen  
Zongkai Lin  
Jean-Paul A. Barthès  
Tangqiu Li (Eds.)

LNCS 3168

# Computer Supported Cooperative Work in Design I

8th International Conference, CSCWD 2004  
Xiamen, China, May 2004  
Revised Selected Papers



Springer

*Commenced Publication in 1973*

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

## Editorial Board

David Hutchison

*Lancaster University, UK*

Takeo Kanade

*Carnegie Mellon University, Pittsburgh, PA, USA*

Josef Kittler

*University of Surrey, Guildford, UK*

Jon M. Kleinberg

*Cornell University, Ithaca, NY, USA*

Friedemann Mattern

*ETH Zurich, Switzerland*

John C. Mitchell

*Stanford University, CA, USA*

Moni Naor

*Weizmann Institute of Science, Rehovot, Israel*

Oscar Nierstrasz

*University of Bern, Switzerland*

C. Pandu Rangan

*Indian Institute of Technology, Madras, India*

Bernhard Steffen

*University of Dortmund, Germany*

Madhu Sudan

*Massachusetts Institute of Technology, MA, USA*

Demetri Terzopoulos

*New York University, NY, USA*

Doug Tygar

*University of California, Berkeley, CA, USA*

Moshe Y. Vardi

*Rice University, Houston, TX, USA*

Gerhard Weikum

*Max-Planck Institute of Computer Science, Saarbruecken, Germany*

Weiming Shen Zongkai Lin  
Jean-Paul A. Barthès Tangqiu Li (Eds.)

# Computer Supported Cooperative Work in Design I

8th International Conference, CSCWD 2004  
Xiamen, China, May 26-28, 2004  
Revised Selected Papers

Volume Editors

Weiming Shen  
National Research Council Canada – IMTI  
800 Collip Circle, London, Ontario, Canada N6G 4X8  
E-mail: weiming.shen@nrc.gc.ca

Zongkai Lin  
Chinese Academy of Sciences  
Institute of Computing Technology  
100080 Beijing, China  
E-mail: lzk@ict.ac.cn

Jean-Paul A. Barthès  
Université de Technologie de Compiègne  
Centre de Recherches de Royallieu  
60205 Compiègne, France  
E-mail: barthes@utc.fr

Tangqiu Li  
Xiamen University  
Department of Computer Science  
361005 Xiamen, China  
E-mail: tqli@xmu.edu.cn

Library of Congress Control Number: 2005934226

CR Subject Classification (1998): H.5.3, H.5.2, H.5, H.4, C.2.4, D.2.12, J.6, D.4, H.2.8

ISSN 0302-9743  
ISBN-10 3-540-29400-7 Springer Berlin Heidelberg New York  
ISBN-13 978-3-540-29400-9 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

[springeronline.com](http://springeronline.com)

© Springer-Verlag Berlin Heidelberg 2005  
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India  
Printed on acid-free paper SPIN: 11568421 06/3142 5 4 3 2 1 0

## Preface

The design of complex artifacts and systems requires the cooperation of multidisciplinary design teams using multiple commercial and non-commercial engineering tools such as CAD tools, modeling, simulation and optimization software, engineering databases, and knowledge-based systems. Individuals or individual groups of multidisciplinary design teams usually work in parallel and separately with various engineering tools, which are located on different sites, often for quite a long time. At any moment, individual members may be working on different versions of a design or viewing the design from various perspectives, at different levels of detail.

In order to meet these requirements, it is necessary to have effective and efficient collaborative design environments. These environments should not only automate individual tasks, in the manner of traditional computer-aided engineering tools, but also enable individual members to share information, collaborate and coordinate their activities within the context of a design project. CSCW (computer-supported cooperative work) in design is concerned with the development of such environments.

A series of international workshops and conferences on CSCW in design started in 1996. The primary goal of these workshops/conferences is to provide a forum for the latest ideas and results on the theories and applications of CSCW in design, research of multi-agent systems, Internet- and Web- based applications, electronic commerce and other related topics. It also aims at promoting international scientific information exchange among scholars, experts, researchers and developers in the field of CSCW in design. The major topics of CSCWD workshops/conferences include:

- techniques, methods, and tools for CSCW in design;
- social organization of the computer-supported cooperative process;
- knowledge-intensive cooperative design;
- intelligent agents and multi-agent systems for cooperative design;
- workflows for cooperative design;
- VR technologies for cooperative design;
- Internet/Web and CSCW in design;
- Grids, Web services and Semantic Web for CSCW in design;
- CSCW in design and manufacturing
- cooperation in virtual enterprises and e-businesses;
- distance learning/training related to design;
- applications and testbeds.

Since this is the first book on CSCW in design in the series of Lecture Notes in Computer Science (LNCS), we would like to provide a little more background information about the history of the CSCWD workshops/conferences. The University of Technology of Compiègne in France and the Institute of Computing Technology of the Chinese Academy of Sciences started an international collaborative project in the area of CSCW in design in 1993. Based on this collaboration, the 1<sup>st</sup> International Workshop on CSCW in design (CSCWD 1996) was held on May 8–11, 1996 in Beijing, China and the second one (CSCWD 1997) was held on November 26–28, 1997, in Bangkok, Thailand. After the two successful workshops, an international working group on CSCW in Design was created and an International Steering

Committee was formed in 1998 (<http://www.cscwid.org>). The Steering Committee then coordinated two workshops (CSCWD 1998 on July 15–18, 1998 in Tokyo, Japan and CSCWD 1999 on September 29–October 1, 1999 in Compiègne, France). During the annual Steering Committee meeting held at CSCWD 1999, the International Steering Committee decided to change the name from the “International Workshop on CSCW in Design” to the “International Conference on CSCW in Design”. The 5<sup>th</sup> International Conference on CSCW in Design (CSCWD 2000) was then held on November 29–December 1, 2000 in Hong Kong, China, followed by CSCWD 2001 on July 12–14, 2001 in London, Ontario, Canada and CSCWD 2002 on September 25–27, 2002 in Rio de Janeiro, Brazil.

The 8<sup>th</sup> International Conference on CSCW in Design (CSCWD 2003) was scheduled to be held on October 22–24, 2003 in Xiamen, China. Due to the outbreak of SARS early in 2003, the conference was rescheduled for May 26–28, 2004 (as CSCWD 2004). Two volumes of conference proceedings were published: Volume 1 in 2003 with 134 papers selected from 170 submissions and Volume 2 in 2004 with 148 papers selected from 188 submissions. This book includes 45 articles that are the expanded versions of the papers presented at CSCWD 2004.

Many people contributed to the preparation and organization of CSCWD 2003 / CSCWD 2004. We would like to thank all Program Committee members for their efforts in promoting the conference and carefully reviewing the submitted papers, as well as the authors who contributed to the conference.

We would also like to thank the chairs and members of the Organizing Committee for taking care of all the details that made CSCWD 2004 successful. We acknowledge the sponsorship of Xiamen University, China and the co-sponsorship of the IEEE Beijing Section, the CIMS Committee of the National Hi-Tech R&D Program of China, the China Computer Federation, the National Natural Science Foundation of China, Zhongshan University, China, and Fuzhou University, China.

Special thanks to Prof. Wenhua Zeng, Prof. Shaozhi Li, Prof. Chenhui Yang, Zhongpan Qiu, Dandan Liu, Youzhun Xu, Xinzhen Xu, and Xiaosu Zhan who made significant contributions to the preparation of the conference and the editing of the conference proceedings.

July 2005

Weiming Shen  
Zongkai Lin  
Jean-Paul Barthès  
Tangqiu Li

# Table of Contents

## CSCW Techniques and Methods

Vega Information Grid for Collaborative Computing <i>Zhiwei Xu, Ning Yang, Huaming Liao</i> .....	1
Physical Object Icons Buttons Gesture (PIBG): A New Interaction Paradigm with Pen <i>Guozhong Dai, Hui Wang</i> .....	11
A Novel Method of QoS Based Resource Management and Trust Based Task Scheduling <i>Junzhou Luo, Peng Ji, Xiaozhi Wang, Ye Zhu</i> .....	21
Learning to Plan the Collaborative Design Process <i>Flávia Maria Santoro, Marcos R.S. Borges, Neide Santos</i> .....	33
Groupware System Design and the Context Concept <i>Marcos R.S. Borges, Patrick Brézillon, Jose Alberto Pino, J.-Ch. Pomerol</i> .....	45
Grid Authorization Management Oriented to Large-Scale Collaborative Computing <i>Changqin Huang, Zhiting Zhu, Xianqing Wang, Deren Chen</i> .....	55
Research on Network Performance Measurement Based on SNMP <i>Shufen Liu, Xinjia Zhang, Zhilin Yao</i> .....	67
Concepts, Model and Framework of Cooperative Software Engineering <i>Yong Tang, Yan Pan, Lu Liang, Hui Ma, Na Tang</i> .....	76
An Algorithm for Cooperative Learning of Bayesian Network Structure from Data <i>Jiejun Huang, Heping Pan, Youchuan Wan</i> .....	86
Non-violative User Profiling Approach for Website Design Improvement <i>Jiu Jun Chen, Ji Gao, Song En Sheng</i> .....	95

## Agents and Multi-agent Systems

Generative Design in an Agent Based Collaborative Design System <i>Hong Liu, Liping Gao, Xiyu Liu</i> .....	105
--	-----

Similarity Based Agents for Design  
*Daniel Pinho, Adriana Vivacqua, Sérgio Palma, Jano M. de Souza . . .* 117

Semantic Integration in Distributed Multidisciplinary Design  
 Optimization Environments  
*Ying Daisy Wang, Weiming Shen, Hamada Ghenniwa . . . . .* 127

Formal Dialogue and Its Application to Team Formation in Cooperative  
 Design  
*Yisheng An, Renhou Li . . . . .* 137

MA\_CORBA: A Mobile Agent System Architecture Based on CORBA  
*Xingchen Heng, Chaozhen Guo, Jia Wu . . . . .* 147

A Multi-agent Based Method for Handling Exceptions in Computer  
 Supported Cooperative Design  
*Feng Tian, Renhou Li, M.D. Abdulrahman, Jincheng Zhang . . . . .* 156

**Ontology and Knowledge Management**

CEJ – An Environment for Flexible Definition and Execution  
 of Scientific Publication Processes  
*Daniel S. Schneider, Jano M. de Souza, Sergio P. Medeiros,  
 Geraldo B. Xexéo . . . . .* 165

Methodology of Integrated Knowledge Management in Lifecycle  
 of Product Development Process and Its Implementation  
*Peisi Zhong, Dazhi Liu, Mei Liu, Shuhui Ding, Zhaoyang Sun . . . . .* 175

Knowledge-Based Cooperative Design Technology of Networked  
 Manufacturing  
*Linfu Sun . . . . .* 187

Multi-ontology Based System for Distributed Configuration  
*Xiangjun Fu, Shanping Li . . . . .* 199

**Collaborative Design and Manufacturing, and  
 Enterprise Collaboration**

Online Collaborative Design Within a Web-Enabled Environment  
*Daizhong Su, Jiansheng Li, Shuyan Ji . . . . .* 211



C-Superman: A Web-Based Synchronous Collaborative CAD/CAM System <i>Weiwei Liu, Laishui Zhou, Haijun Zhuang</i> .....	221
Developing a Multidisciplinary Approach of Concurrent Engineering <i>Heming Zhang, David Chen</i> .....	230
Hardware/Software Co-design Environment for Hierarchical Platform-Based Design <i>Zhihui Xiong, Sikun Li, Jihua Chen, Maojun Zhang</i> .....	242
A Computer Supported Collaborative Dynamic Measurement System <i>Peng Gong, Dongping Shi, Hui Li, Hai Cao, Zongkai Lin</i> .....	252
A Collaborative Management and Training Model for Smart Switching System <i>Xiaoping Liao, Xinfang Zhang, Jian Miao</i> .....	260
A Web-Based Fuzzy-AHP Method for VE Partner Selection and Evaluation <i>Jian Cao, Feng Ye, Gengui Zhou</i> .....	270
A Method of Network Simplification in a 4PL System <i>He Zhang, Xiu Li, Wenhuan Liu</i> .....	279
<b>Virtual Reality and Applications</b>	
Using Augmented Reality Technology to Support the Automobile Development <i>Jürgen Fründ, Jürgen Gausemeier, Carsten Matysczok, Rafael Radkowski</i> .....	289
Real-Time Selective Scene Transfer <i>Min Tang, Zheng-ming Ying, Shang-ching Chou, Jin-xiang Dong</i> .....	299
Design and Implementation of a Collaborative Virtual Shopping System <i>Lu Ye, Bing Xu, Qingge Ji, Zhigeng Pan, Hongwei Yang</i> .....	309
Digital Virtual Human Based Distance Education System <i>Liyun Liu, Shaorong Wang, Fucang Jia, Hua Li, Zongkai Lin</i> .....	319

## Workflows

Towards Incompletely Specified Process Support in SwinDeW – A Peer-to-Peer Based Workflow System <i>Jun Yan, Yun Yang, Gitesh K. Raikundalia</i> .....	328
A Flexible Workflow Model Supporting Dynamic Selection <i>Shijun Liu, Xiangru Meng, Bin Gong, Hui Xiang</i> .....	339
Temporal Logic Based Workflow Service Modeling and Its Application <i>Huadong Ma</i> .....	349
Research on Cooperative Workflow Management Systems <i>Lizhen Cui, Haiyang Wang</i> .....	359
Effective Elements of Integrated Software Development Process Supported Platform <i>Min Fang, Jing Ying, Minghui Wu</i> .....	368

## Other Related Approaches and Applications

Hierarchical Timed Colored Petri Nets Based Product Development Process Modeling <i>Hong-Zhong Huang, Xu Zu</i> .....	378
An Intelligent Petri Nets Model Based on Competitive Neural Network <i>Xiao-Qiang Wu</i> .....	388
An Automatic Coverage Analysis for SystemC Using UML and Aspect-Oriented Technology <i>Yan Chen, Xuan Du, Xuegong Zhou, Chenglian Peng</i> .....	398
Optimistic Locking Concurrency Control Scheme for Collaborative Editing System Based on Relative Position <i>Qirong Mao, Yongzhao Zhan, Jinfeng Wang</i> .....	406
Research on Content-Based Text Retrieval and Collaborative Filtering in Hybrid Peer-to-Peer Networks <i>Shaozi Li, Changle Zhou, Huowang Chen</i> .....	417
On the Stochastic Overlay Simulation Network <i>Ke-Jian Liu, Zhen-Wei Yu, Zhong-Qing Cheng</i> .....	427

Applying Semiotic Analysis to the Design and Modeling of Distributed Multimedia Systems <i>Mangtang Chan, Kecheng Liu</i> .....	437
A Rapid Inducing Solid Model Towards Web-Based Interactive Design <i>Hongming Cai, Yuanjun He, Yong Wu</i> .....	448
<b>Author Index</b> .....	457

# Vega Information Grid for Collaborative Computing

Zhiwei Xu, Ning Yang, and Huaming Liao

Institute of Computing Technology, Chinese Academy of Sciences,  
Beijing, 100080, P.R. China  
z xu@ict.ac.cn

**Abstract.** This paper looks at computer supported cooperative work (CSCW) from an information grid viewpoint. We illustrate two collaborative instances in information grid and point out new CSCW requirements. We discuss key pieces of two models in the VEGA-IG (VEGA information grid): model of object and that of subject, which help the achievement of collaborative work and partly solve the difficulties in collaborative work such as unknown participant, dynamic cooperation channel, multi-modal communication, dynamic participants and resources. The two models take loose coupling as their main point and shield many complicated information of the collaborative work. We also discuss two examples of collaborative work in VEGA-IG.

## 1 Introduction

Research in Computer Supported Cooperative Work (also known as CSCW, groupware, collaboration tools) can be traced as far back as 1968, when Douglas Engelbart demonstrated his NLS with video teleconferencing features [1]. Since then, CSCW has blossomed into an exciting academic discipline with a large, growing application market. ACM now sponsors an annual international conference on CSCW. There are many industrial strength software products. More importantly, we are witnessing an accelerating trend of CSCW applications with ever increasing richness and diversity, ranging from business workflow to multiplayer games.

The term “grid” has been used in various contexts with different meanings. In this paper, we use the broad definition [2] to refer grid to an interconnected distributed system that supports resource sharing, collaboration, and integration. Grid computing [3] is a kind of distributed supercomputing, in which geographically distributed computational and resources are coordinated for solving problems [4]. In this paper, we examine CSCW from an information grid viewpoint. We first give two examples in the real world to summarize the new requirements of the collaboration work. And also we can see that information grid commit itself to the research on enabling technology for information sharing, information management, and information services in grid. We point out that object and subject are important concepts in the research on information grid. The research on the model of object and subject in VEGA-IG help to finish the collaborative work especially with the help of the virtual layer and the effective layer. Information technology as a whole is entering a mass adoption stage, with network computing a main technology characteristic. This trend will have profound implications for the CSCW community. The information grid research community is developing new technology to meet these requirements.

The rest of the paper is organized as follows: In Section 2, we propose two examples in current information grid and discuss the new requirements for CSCW technology. In Section 3, we describe two models of VEGA-IG: the model of object and that of subject. We also discuss the central features of the two models that are relevant to CSCW. In Section 4, we illustrate how the two models help the achievement of collaborative work. We offer concluding remarks in Section 5.

## 2 New Requirements for Collaboration

To see the future requirements of CSCW, we can look at examples in current information systems. For instance, China Railways has the largest customer base in the world, serving hundreds of millions of customers daily. We discuss below two desired collaboration use cases from China Railways information systems. The first case is for business professionals, while the second concerns end users.

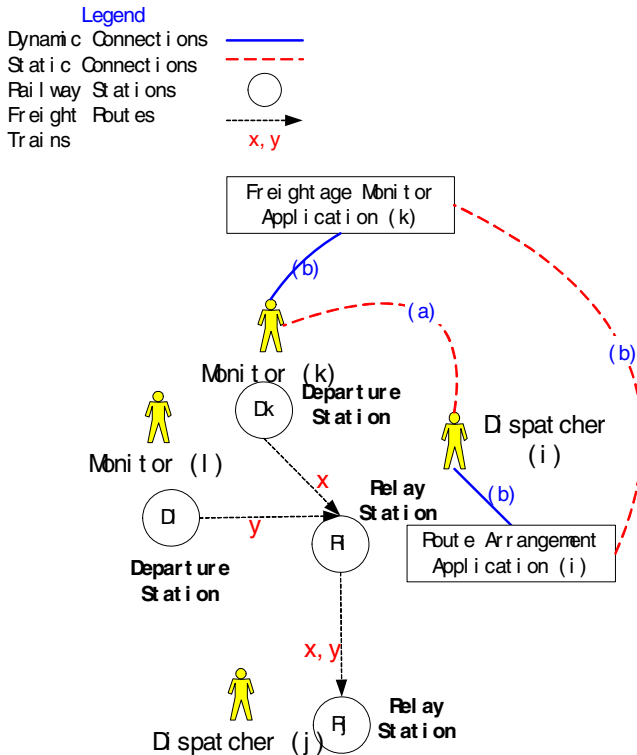


Fig. 1. Freight Route Consultation

**Collaboration in Freight Routing.** Every day thousands of container cargos are transported by railways. Cargoes are composed into freight trains at each departure station. When a freight train passes by a relay station, its route is often adjusted in

accordance with the actual railway situation. The dispatcher in the relay station needs a comprehensive collaboration system help he or she to consult with the train's monitor to decide a reasonable route of the train.

At the same time, related applications need to be integrated together. Figure 1 shows two of such applications. When adjusting a train's route, the Route Arrangement Application must dynamically connect to the train's Freight Monitor Application to exchange information.

Hence they need build two kinds of cooperation channel. One of them is direct communication way through some audio or video conversation medium, denoted as (a) in Figure 1, whereas another one is an indirect communication way through related application(s), denoted as (b) in Figure 1. That implies the information from one participant will be processed by the related applications, and another participant can only observed the processing results.

In our scenario, because it is impossible to predict which train has to change its route in advance, the cooperation channel has to be constructed dynamically and temporarily. This will introduce many new challenges to collaboration system development.

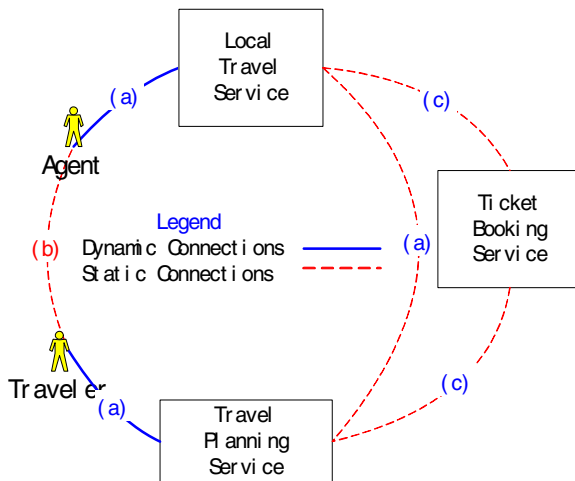


Fig. 2. Collaboration in Traveling

**Collaboration in Traveling.** Traveling by trains is cheap and more and more convenient with the raising quality of railway service. So in China, most common people would choose to travel by trains. Many travelers like to arrange their travel plan well, before set out on a trip. But when they discover some interesting travel program temporarily, they often want to get some service from a local travel agent. At that time, they require collaboration between their original travel service, local travel service, and even ticket booking service.

Consider the collaboration scenario of Figure 2. The traveler needs to look for a local travel service agent. In order to modify the travel plan, the traveler needs communicate with the agent. They could directly communicate through a simple chat

system. In general, the traveler hopes to inherit his original plan, so the related information should be transferred from the travel planning service to the local travel service and get an appropriate new plan. And in the case to change the old travel route, or build a new travel route, the travel planning service, sometimes as well as the local travel service, need collaborate with a ticket booking service automatically. Because the related applications are not aware of each other in advance, it will introduce new challenges to perform the cooperative tasks dynamically.

From these two use cases, we can summarize several collaboration requirements.

**Unknown Participant.** In general, before a cooperative work begins, the collaborative initiator either is aware of the location of other participants (such as message exchange system), or can create a persistent collaborative channel (such as bulletin board system). But in a dynamic collaboration environment, the above assumptions may be broken. Because the collaborative initiator often need initiate collaboration without awareness of the concrete location of other participants, he just knows some rules to look for the other participants. So he could not create a collaboration channel in advance.

For example, in the Freight Route Consultation case, when some trains need to change their route, the dispatcher only knows the rules to look for the train's monitors, but he does not know where they are and how to contact with them. And the issues are similar in the second case.

**Dynamic Cooperation Channel.** After the initiator finds the other participants, to create cooperation channels dynamically will introduce other challenges. Let us look at the first use case. The resulting data must be transferred from the Route Arrangement Application to The Freight Monitor Application. This in turn requires the applications to understand the syntax and semantics of each other's data formats, as well as each other's interface.

Moreover the dispatcher and the monitor also have the possibilities to communicate though different conversation application clients. They need exchange information between the different conversation applications. But, in fact, the applications do not know each other in advance, and at most times they do not want to know so many details of each other. So the problem is how to exchange information among the related applications without knowing access details.

**Multi-modal Communication.** Each collaboration session may need different types of communication. It is desirable if all these communication modes are made available at the user's fingertip.

**Dynamic Participants and Resources.** There are more than one participant and many resources in one collaborative work usually. Much information of both subject and object may have some changes during the period of collaborative work happen. We can take the subject as an example. Users may change their attributes; their host community at any moment, and also the administrator may change some policy of user management. At this moment we must guarantee the accomplishment of the work.

### 3 Two Models in Vega Information Grid

We should take two important factors into account when we refer to CSCW: subject and object. From the viewpoint of philosophy, subject means the people who have the ability to cognize something and practice, while object means those things that existed besides subject in the world. To be brief we can consider that in VEGA-IG, the object is the resource and the subject is the user. As it comes to collaborative work, there should be at least two participants (users namely) who are involved in the work. And at the same work, there should be some resources that act as object. Only when all the participants work at the same resource can the collaborative work achieved. Resource sharing is also the primary method supports the collaborative work. So it is safe for us to draw the conclusion that the research of subject and object in VEGA-IG will be beneficial to that of CSCW.

We propose the object model and subject model of VEGA-IG respectively in this section.

#### 3.1 EVP Model of Object in VEGA-IG

Firstly we discuss some points in the object model of VEGA-IG. The structure of the model is illustrated in Figure 3 [5]. There are three layers in the model named physical layer, virtual layer and effective layer from bottom-up point of view.

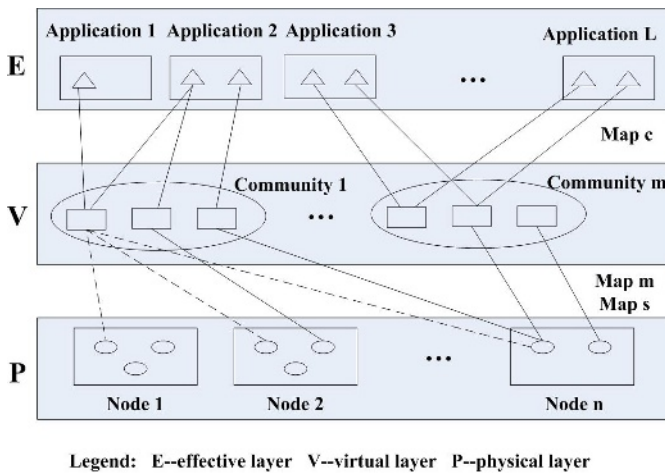


Fig. 3. EVP model of object in VEGA-IG

Physical layer is the lowest layer in the model and is actually where the resource saved and the operation of resource (such as add, delete, modify etc.) happened. The programmers and the professional mainly use resource of this layer. What really happens to the resource of this layer is unknowable to the end users.

Virtual layer has the responsibility of managing the resources in a community and dynamically adjust the map between the virtual layer and the physical layer when



some kind of alteration happen. With the help of virtual layer resource, many collaborative works can be accomplished conveniently for this layer help users to locate and search the resource he or she need.

End user uses effective layer and the resource provided by this layer is available even for the secretary. A higher level language called Grid Service Markup Language (GSML) is available in this layer, which allows users (not necessarily programmers) to specify grid services and user interface in an easy to use fashion. GSML makes collaborative work among mass users possible.

Resources on grid nodes can be wrapped as Web services or grid services, and registered with the Vega GOS (Grid Operating System). Resources at this level belong to physical resources. A technical staff can use a software tool called Resource Mapper to map physical resources into location-independent virtual resources. A secretary can then use another tool called GSML composer to turn the virtual resources into effective resources in a GSML page. Table 1 gives some main features of the object model.

**Table 1.** Features of Object EVP Model in VEGA-IG

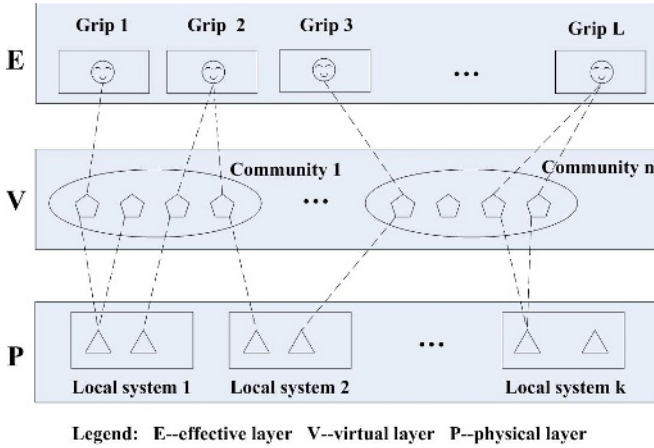
Layer	Program language	Developer	User	Resource assemble
Effective	High level	Secretary	Leader	Yes
Virtual	Middle level	Programmer	Secretary	Congener resource
Physical	Low level	Programmer	Programmer	No

### 3.2 EVP Model of Subject in VEGA-IG

Secondly, we bring forward the EVP space model of subject in VEGA information grid showed in Figure 4 according to the EVP space model of object. There are still three layers named physical layer, virtual layer and effective layer respectively. While what different from the model of object is that security instead of resource is the main point in the model. We don't treat the information of subject same as that of object. Besides one kind of resource in the information grid, the information of subject should also have the liability to be the representation of the user.

Effective layer user is mainly designed for the end user and composed of exterior user name and interior user name when it comes to implement. The exterior user name is friendly for the user to logon the grid and support the single sign on (SSO) in the information grid. In the collaborative work where one participant needs to know where and who are the other participants, the user information of this layer can do some help. What really matters in effective layer is that one user (or a grid service, a web service, an application who act as the role of subject) can easily find his or her participants in the information grid.

Virtual layer user is designed to finish such kind of works related with resource and access control (AC). It is composed of certificateID Proxy (a transformed certificateID by some algorithm in order to protect the real certificateID) and the Token where many information of resource and access control saved.



**Fig. 4.** EVP Model of Subject in VEGA-IG

Physical layer user is namely the local user, and the actual user who represents the grid subject to use resource. Physical layer user has much with the system who provides the resource. We can take Data Base as an example where the physical layer user maybe identified as Dbuser @ dbpasswd.

**Table 2.** The Features of Subject EVP Model in VEGA-IG

Layer	User Identify	Feature	Function
Effective	UserID&PassWrod	Exterior name Interior name	Friendly Uniquely SSO
	Certificate ID		
Virtual	Certificate ID Proxy	Community collaborated Resource collaborated	Secretly Flexible Unique in community
	Token		
Physical	Local user	Local system collaborated	Unique in local system

The three-layer partition of the information subject makes many collaborative works possible. Many features of the model is illustrate in Table 2. In the work of fee count system, it is impossible for the system that provides the resource service to know which user indeed use the service since many information grid users may be mapped into the same physical user. While with the help of virtual layer user, it is convenient for the system to distinguish the users who use the service. So it is safe to draw the conclusion that the three-layer partition of the information subject guarantees the security of the system, the character of unique, the SSO, and also the alterability of the policies.

## 4 The Vega-IG and Its Supports for Collaboration

With two models above, we can solve part of the problems exist in the two cases we propose in section 2. For the first case of freight route consultation, the two applications can operate on the same physical layer resource by operations on different virtual layer resource. The monitor can modify the physical layer resource by his own application when some emergences happen while the dispatcher can get the modified information as soon as the change happen. With the help of the virtual layer, there seemed to be a dynamic channel between the two applications. For the second case of collaboration in traveling, because both subject and object are well-organized in community of effective layer or (and) virtual layer, the traveler can easily find the services he or she want by many kind of tools such as Service Search. With the help of Service Search, the traveler can find those services that satisfy his or her need and then establish a dynamic channel between the two services.

From analysis above we can see that Information grid commit itself to the research on enabling technology for information sharing, information management, and information services in grid. The information grid focuses on information, and largely ignores issues such as process, transaction, and semantics. VEGA-IG has four features as follows:

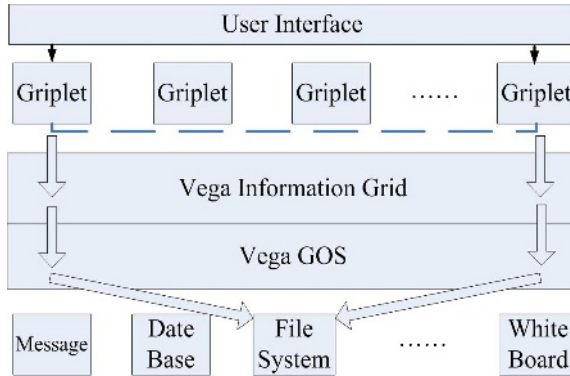
- 1) Platform character
- 2) Dynamic adaptability
- 3) Sharing character
- 4) Collaborative character

The infrastructure of VEGA information grid is illustrated in Figure 5. Griplet is a kind of Grip (grid process) who represents the subject of grid to visit and use grid. Grip is a key concept in the research on other kind of grid such as computing grid. What different from other system is that VEGA information grid gives users a platform with which people can add both resource and applications into the system conveniently. Different kind of resources such as messages, databases, file systems and also white board can be added to the bottom of the platform, and at the same time, different kind of applications can be added to the top of the platform. Using this platform we can implement some kind of collaborative task and the method here is far from the traditional method of “solution”.

Now we take resource transfer in VEGA-IG as an example to show how the collaborative task is accomplished based on the two models. At present, a user can transfer files to another user through the methods of mail, msn etc. All of these methods use the real resource as an attachment. That means the copy of the resource is produced and transferred to another place. While in VEGA information grid, what a user should do is just transfer the handle of the file (identifier of the file in effective layer) to the destination user. Then the receiver can use the handle to locate where the resource is by the maps between the layers. Both of the participants may have no idea about where the resource really is, they don't know the existence of the three layers either. It is obviously that the handle must keep in accordance with the rule or the policy of the model. As soon as the destination user gets the handle, he or she can parse the handle and locate the resource also by the maps between the layers. We can see that when the work happens, there seemed to be a temporary channel links all the

participants through which the resource is transferred. In fact what really transmitted between the two participants is the identifier of the file in effective layer.

In the example above, the two participants are familiar with each other. In this paragraph we give another example where one participant have no idea about any



**Fig. 5.** The Infrastructure of VEGA-IG

things of the other participants such as who, from where they are; how many participants there are and even whether there are other participants in the task. We have implemented the projects manage tools in VEGA information grid. As we know there will be more than one person who participant into a project, the management of the project is a collaborative computing job. When we take AC (access control) into account, we can decide who have the right to manage the project by their roles. All the users who have the right to appraise the project may don't know the existence of others. Obviously they are different users in the effective layer, while when they come to the physical layer, maybe they are all mapped into the same local user who has the ability to modify the resource. Different user can modify different virtual layer resource, the collaborative task can be finished when all the virtual layer resource are mapped into the same physical layer resource. Finally we can use the physical layer resource to make a chart to illustrate the unitary state of the project.

In fact the projects manage tools in VEGA information grid use the technique of resource sharing to gain its ends. The similar collaborative work happened in VEGA-IG are flow, calendar etc.

## 5 Conclusions

As CSCW and information grid technology both aim to supporting collaboration, they are closely related. Information grid technology could provide a general-purpose technology platform for the CSCW community to build and run collaboration applications.

Differing from traditional CSCW software tools that focus on enterprise collaboration, information grid technology aims to provide supports for dynamic, multi-modal collaboration based on open standards.

The Vega Grid project is researching a suite of techniques to better support resource sharing and collaboration. The three-layer object and subject model facilitate usability and flexibility; it also gives sustentation of the dynamic character of grid and partially solves the problems of unknown participant, dynamic cooperation channel, multi-modal communication, dynamic participants and resources.

Collaboration in a grid environment is still a young field. Much remains for research. However, it is critical that all work should be based on open standards. Individual pieces of grid technology are integrated according to a grid architecture framework. Examples include Open Grid Service Architecture (OGSA) [6] and Web Service Architecture (WSA) [7]. Isolationism and protectionism can only hinder the development and the use of grid-based collaboration technology.

## References

1. Hofte, H. t.: Working Apart Together: Foundations for Component Groupware. Telematica Instituut Fundamental Research Series, Vol. 001. Enschede, the Netherlands: Telematica Instituut, (1998)
2. Malone, M.S.: Internet II: Rebooting America. Forbes ASAP, (2001)
3. Foster, I., Kesselman, C. (Eds): The Grid 2: Blueprint for a New Computing Infrastructure, Morgan Kaufmann Publishers, (2004)
4. Li, Ch., Xiao, N., Yang, X.: Selection and Advanced Reservation of Backup Resources for High Availability Service in Computational Grid. The Second International Wrokshop on Grid and Cooperative Computing, 2 (2003) 755-762
5. Li, W., Xu, Z.: A Model of Grid Address Space with Application. Journal of Computer Research and Development, 40(12) (2003) 1756-1762
6. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: Grid Services for Distributed Systems Integration, *IEEE Computer*, 35(6) (2002) 37-46
7. Booth, D., Haas, H., McCabe, F., Newcomer, E., etc.: Web Services Architecture. <http://www.w3c.org/TR/ws-arch>

# Physical Object Icons Buttons Gesture (PIBG): A New Interaction Paradigm with Pen

Guozhong Dai and Hui Wang

Intelligence Engineering Lab,  
Institute of Software, Chinese Academy of Sciences,  
Beijing 100080, P.R. China  
{Dgz, Hui.Wang}@iel.iscas.ac.cn

**Abstract.** Pen-based User Interface (PUI) is becoming more and more popular. This paper presents (the) universal analysis and specification of characters and structure of PUI. PIBG, a new interaction paradigm, is proposed as the framework for PUI. PIBG Toolkit is also developed as a PUI Software Platform. With the goal of high usability, several applications are developed with PIBG Toolkit. PUI provides a method for natural and concordant interaction, which could facilitate the development and applications of collaborative systems.

## 1 Introduction

There will be three stages for the development of computers: Mainframe computers, Desktop PC and Ubiquitous Computing [1]. During the Mainframe computers age, many people use one computer; the interface is text mode, and people input the commands through keyboards. Human Computer Interaction (HCI) is not important because the main purpose of Mainframe computers is to finish the tasks designed before running, which don't need much human's inputs. During the Desktop PC age, each person has his own computer, and Graphical User Interface (GUI) is the interaction mode of PC. Desktop PC has dominated for about two decades. GUI is based on the Desktop metaphor and applies the WIMP (Windows, Icons, Menus, Pointer) paradigm. This interaction style has many obvious advantages such as visualization of interaction objects, minimal syntax, and fast semantic feedback [2]. What will be the HCI of Ubiquitous computing like? Many research works have addressed this question. Invisibility would be the most distinguished character of Ubiquitous computing; as Weiser said, "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." [1]. Good user interfaces should make the users focus on the tasks, not on the interfaces themselves; that is to say good user interfaces are invisible to users. The natural user interfaces don't require users' active attention [1]; all things happen in natural way. The objective of invisible computing is to present a natural, high efficient interaction way to decrease the cognitive distance between human and the computers.

Pen and paper have a long history as a tool for recording information. Pen-Paper metaphor is a universal and fundamental way of capturing daily experience, commu-

nicating ideas, recording important events, conducting critical thinking and visual description. The informal nature of pens allows people to focus on their task without having to worry about precision [2]. PUI promises to have many benefits that pen and paper provide.

Users are bounded in the WIMP interactive mode in most current systems; they have to provide precise information and operate according to the ordered steps of menu and icon. PUI will improve the operation process. Users should be able to maintain their thinking without being broken by excessive switches among selection of menus and operations on buttons and keyboard input of words. The interaction mode based on pen-paper metaphor has been deeply infiltrated in natural user interface. It is natural for users to express themselves to other users and to computers, which can provide the imprecise and quick sketching [2].

PUI allows users to directly perform the intended tasks and operations. It can facilitate the development and applications of collaborative systems. The development team members located in different places can share and exchange the production information via network. Since the collaborative design and production in distribution could achieve dynamic union and global cooperation among developers, and the full utilization of global resources, it can greatly reduce the development cycle time of products as well as their costs, and improve the ability to develop personalized products.

The remainder of the paper is organized as follows: Section 2 discusses related work about PUI; Section 3 describes the analysis of PUI; Section 4 presents the proposed framework of PUI; Section 5 gives a brief introduction of the application systems with PUI; Section 6 shows the advantages of collaborative systems with PUI; and finally, Section 7 provides conclusions and discusses the future work.

## 2 Related Work

PUI is becoming more and more popular, versatile, and powerful. This section discusses relevant prior work. The first subsection discusses prior work on PUI. The second subsection gives some background that provides the foundation for this work.

In addition to research and commercial work on handwriting recognition, much work has been done on efficient text input methods and gesture recognition. Many systems use pen-based sketching interfaces to encourage creative activities: SILK [4] uses it for GUI design, MusicPad [5] uses it for music composition, SKETCH [6] and Teddy [7] use it for 3D modeling. Pen-based techniques are commonly used on electronic board systems, with specialized interfaces designed for large boards. For example, a series of papers on the Tivoli [8] proposed many interaction techniques to organize handwritten notes in meeting environment. Other related systems are: LiveBoard [9], DENIM [10], Cocktail Napkin [11], Flatland [12], Classroom 2000 [13], ASSIST [14], etc.

Although these previous works discuss the interaction techniques and specific applications with pen, there still exist barriers to the efficient computer applications supporting by pen based activities in general. Ideas in ubiquitous computing provide the elicitation for research on PUI. The kinds of devices range from large display devices, PC to mobile devices. This opportunity presents challenge for the

development of pen devices for wider application in terms of the characteristics, such as flexibility, convenience and portability. Combination with the context-awareness presented in ubiquitous computing, the PUI provides some kind of invisible interaction to improve the human-computer interaction on the base of universal application. It is important to exchange information fluently and efficiently in the common life by the popular tool. PUI is one of the main styles in Post-WIMP user interfaces.

### 3 Natural User Interface and Pen-Based User Interface

Natural User Interface is one of the important aspects in ubiquitous computing. In the next generation of user interfaces, pen-based input, gesture, speech, perceptual UI, tangible UI will be integrated into human life for interacting with the computers with better services. The proliferation of devices gets the parallel improvement with the corresponding computing technology. Interaction methods can be made more effective when they are adapted to the skills of the designers. Then the designers do not need to pay much attention to the operations of interaction, but the contents of the interaction. One starts to study with speech or pen and paper since he or she was born. It is a great advantage if designers can, for the purposes of HCI, use the interaction skills they have already mastered before. Connecting the natural tools with the powerful processing ability of computers can prevent designers from having to use special methods that distract their attention from the main tasks at hand.

In traditional interaction, users are confined within one eye and one finger for the information exchange. In WIMP interaction, there are many overlapping windows for the virtual information. Users have to select commands through many icons and menu by clicking in small zones. Comparing with WIMP paradigm, the information presentation style and interaction style have changed in PUI. The advantages of PUI are:

- Physical attributes: easy steering, portable and consistence
- Logic attributes: abstract, continuous
- Psychological attributes: focus, intuitive, creative thinking undisrupted

In some conditions, a pen can be used to take the place of mouse, but what fits in with a mouse is not similar with a pen, and vice versa. For any given task, the ideal PUI should not be limited to technique developed for GUI, but incorporate pen-specific techniques that take advantage of the special characteristics of pen. We give some main functions of PUI:

- Pen input as ink, Ink as first-citizen class
- Pen input as gestures including: selecting and moving
- Interpreters that act on ink input (sketching/writing)
- Grouping of objects
- Layering of objects
- Time indexing of ink input
- Transformation of ink to cleaned-up objects
- Immediate and deferred processing of ink

Interaction information in PUI has two important characters. First, the information is continuous, instead of discrete events. Therefore, to capture this continuous



interaction, we use the Jacob’s model to specify the information flowing form pen to PUI. Second, the interaction information created by pen is not only the 2D position in the coordination of paper, but also pressure, orientation, time, etc. Hence the structure of integrating various kinds of information should be set up. We apply a hierarchical structure to integrate multi-model information.

In Figure 1, we give the description of PUI. Based on ink computing, some interaction technologies are provided to improve the human-computer interaction of information exchanging, including gesture, context, life-experience, etc.

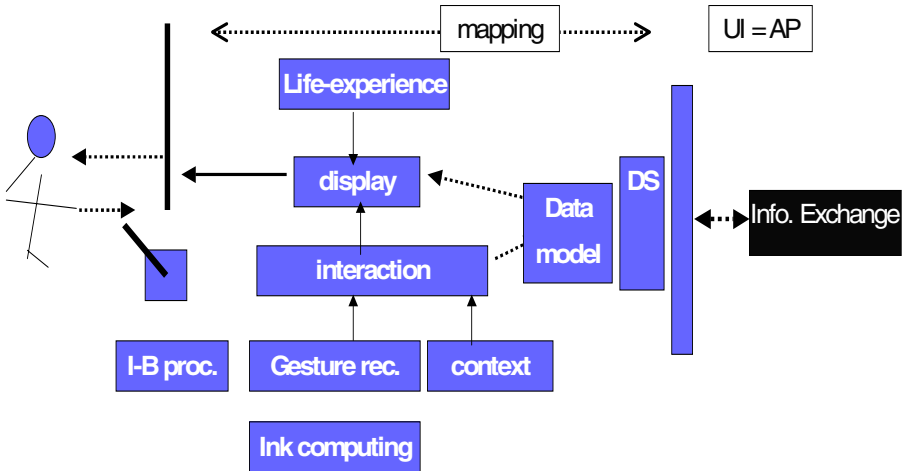


Fig. 1. Pen-based User Interface

#### 4 Pen-Based User Interface Framework

PUI provides freeform operations and structure operations that are different from the WIMP interfaces, as we have discussed in Section 3. The related interaction technologies present higher efficiency and less constraint of operations. From the point of view of cognition psychology, cognition model, user model and semantic model give the specific theory interpretation for the HCI modeling. The PUI framework is given in Figure 2.

Based on the research of information exchanging and framework of the PUI, we present a new interaction paradigm, named PIBG. Two main interaction widgets in PIBG paradigm (paper and frame) belong to Physical Objects. Paper is a kind of widget that serves as a container in PIBG. It has two main responsibilities. First, it receives the information from pen and dispatches it to specific information receiver (a widget in the paper). Second, all widgets contained are grouped and managed as a tree structure by Paper. Frame is the most important widget in the PIBG paradigm. Its responsibilities are processing the interaction information and managing different types of data. Menu is disappeared in PIBG paradigm. In PIBG paradigm, users’ action is changed from mouse pointing to pen gesture [15].

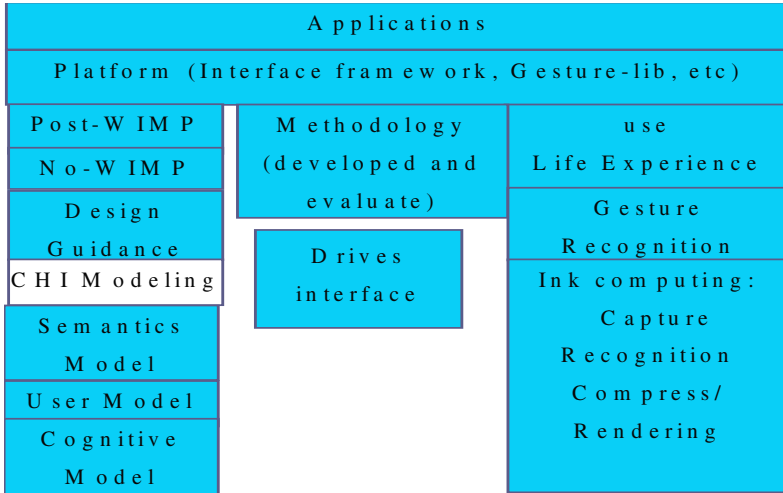


Fig. 2. Framework of Pen-based User Interface

Gestures invoke operations, which provide direct manipulation to objects. However, it is difficult to recognize gestures according to user' intention if the invoked operation information is fuzzy. Users easily detect misrecognition of a character, but misrecognition of an operator may not be. In other words, if a gesture is misrecognized it will cause an unintended operation to be performed, and users may have difficulty in determining what happened. Furthermore, an unintended operation is likely to be more difficult to correct than an incorrectly recognized character. The gesture design model presented above according to the rules based on constraint and the context-awareness can provide better performance [16]. With the development of pen and new interaction technology, gestures are the valuable aspect of PUI. We believe that gesture operation, unlike the majority of WIMP techniques based on mouse motions, has the greatest potential for maximizing the interactivity of pen-based environments since operation modes are offloaded from focusing on the technology to users and tasks themselves. Figure 3 shows some gestures we designed in word processing task. These gestures are designed for the Chinese word processing; and the first gesture is for Inserting, the second one is for Selection, the third one is for Deleting, the fourth one is for Moving, the fifth one is for Inverting and the sixth one is for Replacing.

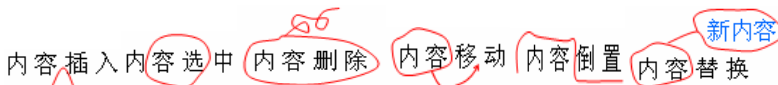


Fig. 3. Gestures in Word Processing Task

The architecture of PIBG paradigm has three new features. It supports continuous interaction information, multi-model interaction information, and probabilistic interaction style. In PIBG, all widgets are built to support Continuous interaction and

multi-model interaction according to the structure. At the same time, we set up a structure that combines recognition, context-aware and User Mediation techniques. Each widget has such structure to support this feature. Figure 4 shows the structure of Paper.

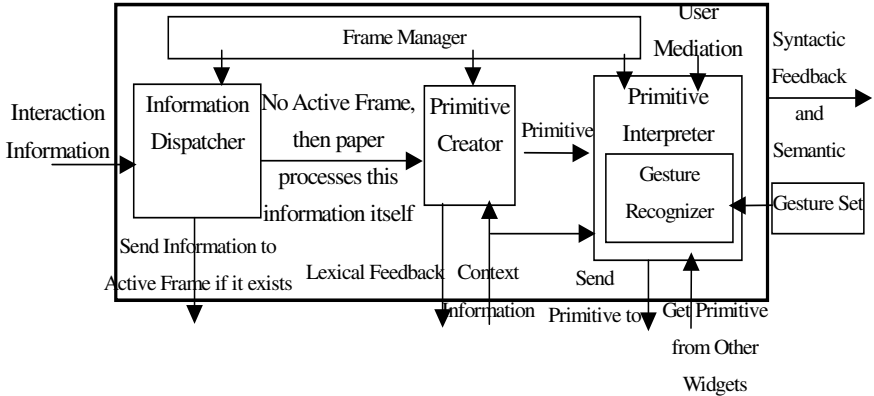


Fig. 4. Structure of Paper

## 5 Applications with Pen-Based User Interface

The real purpose for PUI is the applications. PUI is popular in many kinds of applications. Pen is convenient as the design and writing tool. It can be used to take notes and give annotation to capture and access to life experience. Some killer applications based on pen interaction can be developed as below.

- Browser Production → Consume
- Information access, PIM
- Note-taking (help to capture ideas)
- Collaboration

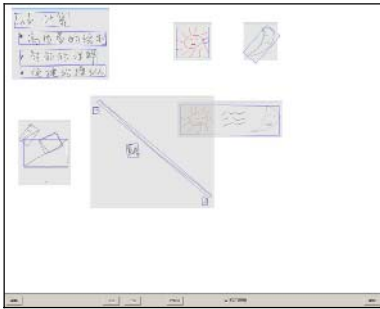
We have built several pen-based interaction systems as shown in Figure 5: (a) PenOffice is a system for teacher preparing their presentations and deliver lectures to a class. (b) State diving team training management system provides a pen input mode for coach to arrange the training plan. (c) Multi-Level Structures Extracting is given to optimize the algorithm of syncopation from Chinese Handwriting. (d) Multimodal geometry is a system of dynamic geometry in terms of multimodal speech and pen. (e) MusicEditor is a system that recognizes handwriting numbered musical notation and exports this information to audio result. (f) A Palette for Children is a sketch system for children easily creating virtual worlds and entities. (g) Learning by pictures system is given for teachers and parents to create pictures for children to learn new words. (h) A virtual play land for children provides the 3D environment for children to learn and play.



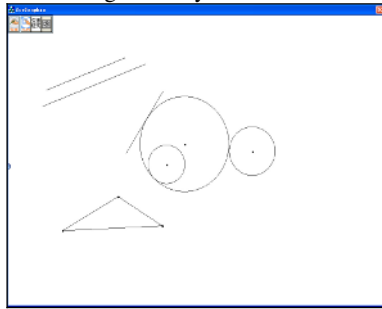
(a) PenOffice



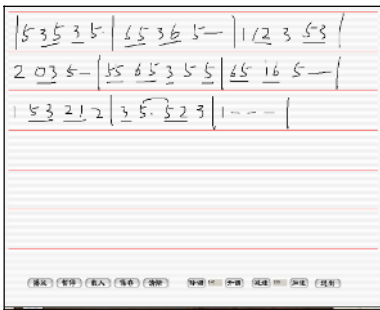
(b) State Diving Team Training Management System



(c) Multi-Level Structures Extracting



(d) Multimodal Geometry



(e) Music Editor



(f) A Palette for Children



(g) Learning by pictures



(h) A virtual play land for children

Fig. 5. Examples of Pen-based Systems

## 6 Collaboration with Pen-Based User Interface

Collaborative designs are necessary in many situations. When creating a collaborative modeling application, there are inherent issues that must be addressed to make the tool intuitive and efficient. A collaborative modeling application's interface must be both fast and direct. Post-WIMP interface metaphor will meet the users' speed and directness that the applications require [17]. Specifically, PUI allows for fast interaction. A gesture-based interface also allows users to directly perform the intended tasks and operations. This directness not only increases interaction speed, but also allows participants to keep their focus on the design session so that they can remain aware of what others are doing in the design environment.

The collaborative design method based on gesture and sketch enhances the design in a natural and concordant way, enabling the share and exchange of sketches and information for the product developers in different places and thus improving the efficiency. It is an efficient and natural tool that conveys thoughts between people and computers as well as people and people. The method proposed in the paper supports multi-view sketch design for designing thinking, as well as collaborative context-awareness. The context based on gesture in pen interaction is an instance of general context in pen interface. And the context-awareness is dynamic, which is concerned with the real-time operation during the interactive process besides the static information. Constraint is an important part in context. The paper presents a context-aware model based on context-aware infrastructure provided by Jason I. Hong at U C Berkeley and hierarchy structure in Schmidt and Dey's model [19]. The contexts such as objects, constraints, and current gestures are static. The contexts, such as users, locations, scopes that are related with specific operations, are dynamic. Generally, it is difficult to provide clear interpretation based on limited information from a sole context. The fusion of different context can provide more information.

One goal of pen interaction is to imitate the operation mode of pen and paper in common life. Users can express themselves under little limitation by use of pen, as is very important to record the transient afflatus during the design process. The gestures are the main operation characters in pen interaction. Commands of free hand stroke gestures issued with pens have semantic mapping applications. For example, gestures such as creation, selection, deletion, modification, constraint confirmation and cancellation, are supported in the system.

The rough sketches drawn by designer serve as a fluent way of expression. Concurrently, the recognized figures can provide more particular and more accurate information. This makes it possible to carry out more detailed designs. Therefore, we should make full use of the flexibility of pen interaction. When recognizing the pen-based sketch, the original track of the sketch should be recorded simultaneously. By providing sketches from two different angles of view, intent of the user can fully exhibited.

## 7 Conclusion

The interaction mode based on pen-paper metaphor is natural for users to express themselves between human and computers. In this paper, we introduce a PUI that

steps away from the rigidity of traditional user interfaces, supporting instead the flexibility and ambiguity inherent in natural modes of communication. How to represent and make the best of the context is still a challenge for us to confront. PIBG Toolkit is a PUI Software Platform built on PIBG Paradigm, which is a new interaction paradigm about PUI. When it is used to develop specific Pen-based applications, the software architecture and interaction control can be constructed automatically; and developers can choose various widgets to build their applications. Therefore, developers can put their focus on specific application domains, instead of things in low-level. PUI provides a method for natural and concordant interaction, which could facilitate the development and applications of collaborative systems. The future research involves the investigation of pen-based interfaces in ubiquitous computing.

## Acknowledgement

This research is supported by the National Key Basic Research and Development Program under Grant No.2002CB312103.

## References

1. Weiser, M.: The Computer for the Twenty-First Century. *Scientific American*, Vol. 265, No. 3 (1991) 94-104
2. Abowd, G.D., Mynatt, E.D.: Charting Past, Present, and Future Research in Ubiquitous Computing. *ACM Transactions on Computer-Human Interaction*, Vol. 7, No. 1 (2000) 29-58
3. Long, A.C., Landay, J.A., Rowe, L.A.: PDA and Gesture Use in Practice: Insights for Designers of Pen-based User Interfaces. Technical Report UCB//CSD-97-976, U.C. Berkeley (1997)
4. Landay, J.A.: SILK: Sketching Interfaces Like Crazy. *Proceedings of Human Factors in Computing Systems (Conference Companion)*, ACM CHI '96. Vancouver, Canada, April 13--18 (1996) 398-399
5. Forsberg, A., Dieterich, M., Zeleznik, R.: The Music Notepad. *Proceedings of the ACM Symposium on User Interface and Software Technology (UIST)*. ACM, ACM Press, New York, NY. USA. Nov. (1998) 203-210
6. Zeleznik, R.C., Herndon, K.P., Hughes, J.F.: SKETCH: An Interface for Sketching 3D Scenes. *ACM SIGGRAPH96 Conference Proceedings*. New Orleans, Louisiana, USA. August 4-9 (1996) 163-170
7. Igarashi, T., Matsuoka, S., Tanaka, H.: Teddy: A Sketching Interface for 3D Freeform Design. *ACM SIGGRAPH99 Conference Proceedings*. Los Angeles, USA, August 8-13 (1999) 409-416
8. Pederson, E.R., McCall, K., Moran, T.P., Halasz, F.G.: Tivoli: An Electronic Whiteboard for Informal Workgroup Meetings. *Proceedings of the InterCHI'93 Conference on Human Factors in Computing Systems*. Amsterdam, the Netherlands. May (1993) 391-398
9. Elrod, S., Bruce, R., Goldberg, D., Halasz, F., Janssen, W., Lee, D., McCall, K., Pedersen, E.R., Pier, K., Tang, T., Welch, B.: LiveBoard: A Large Interactive Display Supporting Group Meetings, Presentations, and Remote Collaboration. *Proceedings of the CHI'92 Conference on Human Factors in Computer Systems*, Monterey, CA, USA, May 3-7 (1992) 599-607

10. Lin, J., Newman, M., Hong, J., Landay, J.: DENIM: Finding a Tighter Fit between Tools and Practice for Web Site Design. *CHI Letters: Human Factors in Computing Systems, CHI '2000*. (2000) 510- 517
11. Gross, M.D.: The Electronic Cocktail Napkin - A Computational Environment for Working with Design Diagrams. *Design Studies*, Volume 17, Issue 1 (1996) 53-69
12. Mynatt, E.D., Igarashi, T., Edwards, W.K., LaMarca, A.: Flatland: New Dimensions in Office Whiteboards. *Proceedings of ACM SIGCHI'99 Human Factors in Computing Systems*, Pittsburgh, PA, USA. May 15-20 (1999) 346-353
13. Abowd, G.D., Atkeson, C.G., Feinstein, A. Hmeto, C., Koope, R., Long, S., Sawhney, N., Tani, M.: Teaching and Learning as Multimedia Authoring: the Classroom 2000 Project. *Proceedings of the fourth ACM international conference on Multimedia*, Boston, Massachusetts, USA. November 18-22 (1996) 187-198
14. Alvarado, C., Davis, R.: Resolving Ambiguities to Create a Natural Sketch Based Interface. *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI-2001*, Seattle, Washington, USA. August 4-10 (2001) 1365-1374
15. Dai, G., Tian, F., Li, J., Qin, Y., Ao X., Wang, X.: Researches on Pen-Based User Interface. *Proceedings of the 10th International Conference on Human-Computer Interaction (HCI International 2003)*, Crete, Greece, June 22- 27 (2003) 1223-1227
16. Ma, C., Dai, G., Teng, D., Chen, Y.: Gesture-Based Interaction Computing in Conceptual Design. *Journal of software*, accepted (in Chinese) (2005)
17. van Dam. A.: Post-WIMP User Interfaces. *Communications of the ACM*, Volume 40, Issue 2 (1997) 63-67
18. Dey, A.K, Abowd, G.D., Salber, D.: A Context-based Infrastructure for Smart Environments. *Proceedings of the First International Workshop on Managing Interactions in Smart Environments (MANSE '99)*, Dublin December (1999) 114-128

# A Novel Method of QoS Based Resource Management and Trust Based Task Scheduling

Junzhou Luo, Peng Ji, Xiaozhi Wang, and Ye Zhu

Department of Computer Science and Engineering, Southeast University,  
210096 Nanjing, P. R. China  
jluo@seu.edu.cn

**Abstract.** In order to get higher efficiency of resource allocation and task scheduling algorithm in grid computing, this paper, based on the analysis of the related work on grid resource management and task scheduling, presents a QoS based structure for Grid Resource Allocation and Management System (GRAM) and a 2-Phase Trust Based Scheduling Algorithm. It is believed that the proposed algorithms can improve the efficiency and reliability of the operation of the grid system.

## 1 Introduction

“A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities...” [1]. The key technologies which affect the grid efficiency involve grid resource allocation & management and task scheduling algorithms. The former mainly implements the effective allocation of computational resources under grid environment to satisfy the task demands, thus to improve the utilization of grid resources. Then based on the highly-efficient resource management and reasonable task division, the latter tracks the execution of the whole task, predicts the adverse factors which may affect the efficiency and takes actions to avoid them (such as task migration), or makes retrieval when computing process delays, and finally accomplishes the task perfectly.

In the Open Grid Services Architecture (OGSA [2]), all resources are organized in a rational way and formed as virtual organizations, which are dynamic and extensible. So the resource management is faced with new challenges. On one hand, in OGSA the grid resources are transparent to grid users, in the form of logical resource. But they are distributed actually and have their own managerial strategies. How to allocate and schedule there resources, and enhance their utilization are still unsettled. On the other hand, different grid services have different QoS requirements on resources, and considering the service cost, different users may have different QoS needs. In OGSA, the QoS characteristics of physical resource cannot represent that of logical resource. How to convert the user's QoS requirements to specific grid QoS parameters becomes a key and challenging issue.

Based on above issues, through analyzing the characteristics of Grid QoS, we propose and set up the layered structure of Grid QoS. On the basis of analyzing the content GRAM based on QoS, we further present the architecture of grid resource



allocation management based on QoS (GRAM-QoS). Through the mapping among different layers of Grid QoS, it converts the user's QoS requirements to specific QoS parameters of resources, and implants the mapping conversion of QoS into the resource selection processes. Considering the characteristics of Grid QoS roundly, GRAM-QoS provides a reasonable model for resource allocation management based on QoS.

This paper first summarizes and analyzes the GRAM systems and task scheduling algorithms in existing models for computational grid, and indicates some problems or deficiencies. Then a novel QoS based Structure for Grid Resource Allocation and Management System and a Trust Based Scheduling Algorithm are presented. Finally we conclude the paper and discuss our future work.

## 2 Related Work and Limitations

The goal of Grid is to utilize all available, free computational resources to overcome difficulties brought by complicated tasks with enormous computing workloads, so how to improve the computational efficiency of the grid becomes the research objective in this field. Since the resource allocation & management and task-scheduling algorithm have direct effect on the grid efficiency, lots of related researches have been made on them.

The structure of the resource discovery and management model depends not only on the computing tasks that need scheduling and the numbers of the resources that need to manage, but also on the types of the domains in which resources are located (single domain or multi-domains). From the view of resource organization, discovery and management mode, researches on grid resource management model are mainly classified as Centralized Model, Distributed Model, Layered Model and Multi-Agent based Model. Although these four models do benefit grid computing a lot, there is still a problem, which is the lack of effective resource discovery and management models to provide intelligent macroscopical means of regulation and control and fast reflection ability to those such as resource information update. In fact, though Centralized, Distributed, Layered and Multi-Agent Based Models provide us many kinds of strategies for resource discovery and management, they have not offer such a kind of merger to the requirements such as intelligent macroscopical regulation and control, rapid response to resource changing, etc.: Centralized Model has more powerful ability for macroscopical regulation and control, but limits the scalability of grid; Distributed Model offers very good support for the scalability, but concentrates too much on the partial, lacking the overall situation view; Layered Model is the compromise of Centralized and Distributed Models, managing and discovering the resource change by tree-structure, but the update information needs to be reported to upper-layer managerial nodes layer by layer till the root, quite difficult to meet the requirement of rapid response; Multi-Agent Based Model collects the change information of resources by means of movable Agent, with a strong scalability and flexibility. However, Multi-Agent Based Model has high requirements on Agent itself and also harsh requirements on the Stub needed by Agent, especially for such a large-scale, worldwide, or unsafe environment -- Grid.

An efficient task-scheduling algorithm can also improve the computational efficiency of the whole grid. As one of the key technologies of grid, it always draws

much attention from domestic and international grid researchers. In numerous task-scheduling algorithms, the thoughts of the following several algorithms, 2-Phase Scheduling Strategy [3], Co-RSPB, Co-RSBF, Co-RSBFP Algorithm Based on Priority and Best Fit Mechanism [4] and Scheduling Algorithm Based on Supply-demand Relationship of the Market [5] are quite novel and have higher reference values. But unfortunately, one problem still exists. That is the lack of effective task scheduling algorithm to combine the high scheduling efficiency, the description of the dynamic characteristics of VO or networks that participate in grid, the elusion to the negative influence brought by the dynamics, and also the QoS requirement of the task presenter. In fact, 2-Phase Scheduling Strategy certainly portrays the dynamics of the network to a certain extent, but because it's limited to two phases (inter-LAN and intro-LAN), it lacks the description of VO with complicated layered structure. It can be extended using VO, but compared with LAN the dynamics of VO is stronger. Co-RSPB has higher overall scheduling benefits, but the high amount of refused requests seriously affects its efficiency; Contrarily, Co-RSBF has a low amount of refused requests, but the low overall benefits make it unlikable to be selected; Co-RSBFP is only a compromise of Co-RSPB and Co-RSBF, whose performance also unsatisfying. Scheduling Algorithm Based on Supply-demand Relationship of the Market utilizes the economics theory to analyze the scheduling strategy, though quite novel, it controls by "price leverage", its convergence may be relatively slow, and not suitable for the fast and changeful application of grid.

### 3 The Logical Structure of QoS-Based GRAM

#### 3.1 Grid QoS

Grid QoS has its unique characteristics because of the technologies adopted by grid. Because grid users only care about QoS of logical resource in VO, we can classify the QoS parameters into five categories in VO layer according to their properties [6]:

- (1) **System QoS** includes resource environment QoS and network QoS, which are defined by the service provider. The two types are both environmental factors and have influence on grid QoS, but they are not the main decisive factors. So, we can abstract and express them as system QoS in VO layer.
- (2) **Logical resource QoS** is used to depict the QoS parameters for logical resource, which are defined by the service provider. It considers synthetically the performance of physical resource, load demands of local task and sharing strategy, etc. It's an abstraction of the performance of physical resource in grid environment and the main decisive factor of grid QoS.
- (3) **Security QoS** includes the parameters about the service security level and the access control strategy offered by the service provider. The service security level QoS is used to meet the user's demands on security. The access control strategy QoS is used to meet the provider's demands on security management. Through the access control strategy QoS, the provider can not only decides the legal user and authorize the seemly right to them, but also maps the security management in grid to native platform. In addition, while abstracting service at multiple levels, it needs to provide the semantic information about the security management through the

access control strategy QoS in order to meet the demands of security QoS negotiate.

- (4) **Reliability QoS** is used to evaluate the reliability of the service information, which is offered and maintained by the service provider and VO together. The service provider can estimate the possible deviation of information according to those uncertain dynamic factors. Based on the estimation, the provider can offer the reliability QoS voluntarily to the VO. In order to prevent the cases that the provider's estimation is not accurate enough and the provider offer fictitious information on purpose, the VO needs to appraise and have final power to make decision about the reliability QoS of the service.
- (5) **Accounting QoS** is used for describing the parameters of the service charge and the correlative management strategy offered by the service provider. In grid, the fee is needed for using the shared resource. Different service providers may have different charge management strategies, or even take different charges to different users. While choosing the service, the user has to consider the cost.

In these five categories of QoS parameters, system QoS and logical resource QoS have important influence on grid QoS, so we can also call them functional QoS parameters. While applying the service, the user can choose different functional QoS parameters, thus he can obtain different QoS results. Security QoS, reliability QoS and Accounting QoS are the descriptions of the service's attribute. They do not have decisive influence on grid QoS. They just offer some necessary QoS information to users. The user cannot determine the specific value of these QoS parameters, so we call them descriptive QoS parameters.

### 3.2 The Layered Structure of Grid QoS

Grid QoS may have different representations to different objects. For example, QoS demands put forward by the end user may be a set of specific QoS parameters or only some simple descriptions such as bad, general, better or best. While the QoS demands on resource is related to logical resource and system, for instance, resources are excellent, system response time  $\leq 180\text{ms}$ , system transmission speeds  $\geq 2\text{ Mb/s}$ , etc. The final QoS parameters are a group of particular numerical values. So the system should be able to map the user's QoS demands to final QoS parameters. Here we can divide grid QoS into layered structure, as Figure 1 shows.

The top of it is the application/grid service layer. In this layer, the service provider should define the specific descriptive QoS parameters such as service security QoS, reliability QoS and accounting QoS. The provider should also define the simplified QoS level, such as bad, general, better, best to meet the user's possible simple QoS demands. In order to meet the user's demands on functional QoS parameter, the service provider should define the different specific system QoS parameters and logical resource QoS parameters. If the service includes some lower-level sub service, the provider should abstract the parameters based on the semantic to ensure the higher-level service has unified semantic QoS definition.

The second is virtual organization layer. The QoS parameters in this layer are that of the upper layer mapped in VO. To descriptive QoS parameters, each sub service can translate them into their own security QoS, reliability QoS and accounting QoS

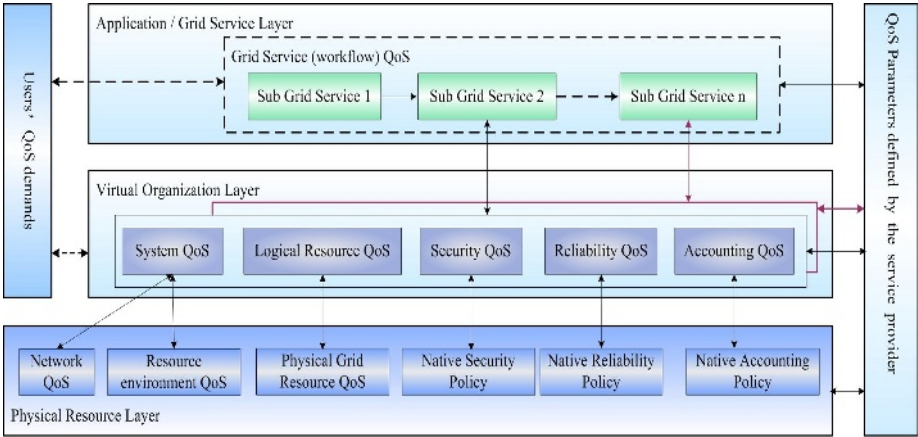


Fig. 1. The Layered structure of Grid QoS

based on their semantic. To functional QoS parameters, the service provider should define each sub service different specific system QoS parameters and logical resource QoS parameters, which corresponded to different simplified QoS levels in upper layer.

The bottom is physical resource layer. The QoS parameters in this layer are that of the translation from VO to native platform. Particularly, system QoS is translated into resource environment QoS and network QoS, logical resource QoS is translated into physical grid resource QoS, security QoS is translated into native security policy, reliability QoS is translated into native reliability policy, accounting QoS is translated into native accounting policy.

Because physical resource is transparent to the user, the user can only pay attention to application/grid service layer and virtual organization layer. In the hierarchical structure model, we express with the broken line that users can choose to put forward QoS demands on different layers according to specific needs. In application/grid service layer, the user can put forward simply QoS demands, for instance bad, generally, better or best etc. In virtual organization layer, the users can put forward specific QoS parameters demands.

Because the relationship of QoS parameters between different layers is defined by the service provider, the provider should define the specific QoS parameters in each layer.

### 3.3 Logical Structure of QoS-Based GRAM

According to the analysis of the layered structure of Grid QoS, we propose the logical structure of QoS-Based GRAM (GRAM-QoS) [7], as Figure 2 shows. The main modules are explained as follows.

**Grid Services Market.** It provides the way to inquire about grid service for the grid user and also to register and publish grid service for the service provider. When the provider registers and publishes, they should provide identity certification and

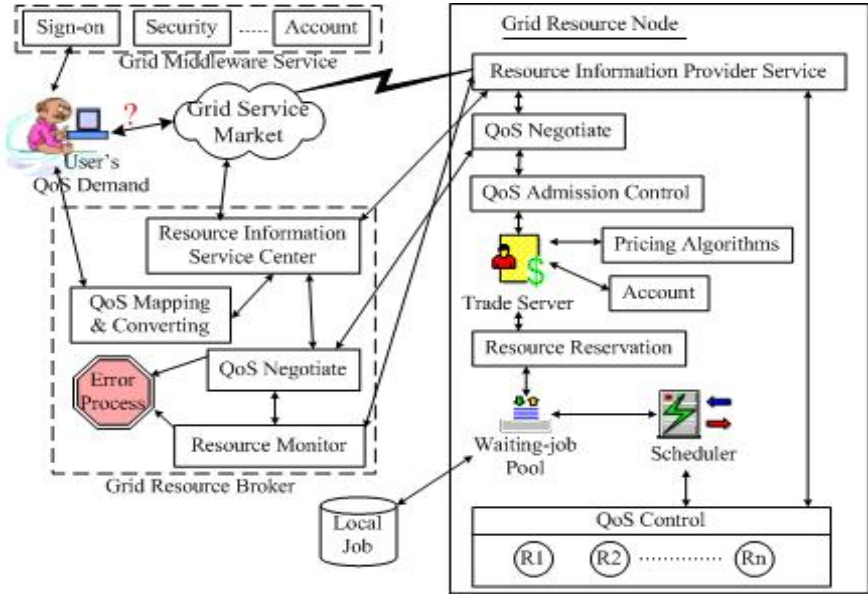


Fig. 2. Logical structure of QoS-Based GRAM

relevant description of service, such as resource demand and QoS demand which with particular QoS parameters in different layers.

**Grid Middleware Services.** This module is mainly responsible for sign-on, safety control, managing user's information and accounting the information about the used resource.

**Grid Resource Broker.** Resource Information Service Center module is the information center of available resource in grid circumstance. It provides information about the quality and QoS parameters of the logical resource. The Resource Information Provider Service module in Grid resource node offers this information. QoS Mapping & Converting module implements the mapping conversion from user's QoS demand to particular QoS parameters in different layers. QoS Negotiation module in Grid Resource Broker is used for judging whether system QoS and logical resource QoS can satisfy user's demands. The QoS Negotiation module in Grid Resource Node judges whether physical resource QoS, network QoS and devices QoS can satisfy the user's demands. When presenting resources cannot satisfy the user's demand, two QoS Negotiation modules should interact with relevant modules and inquire whether the user can reduce QoS demand. Resource Monitor module is responsible for monitoring the reserved resources. If the QoS parameters of reserved resources cannot satisfy user's demands, the module would get touch with the QoS Negotiation module to make new QoS negotiation or choose commutability resource. The Resource Information Provider Service module offers the information needed by this module. Error Process module processes errors that come from the QoS Negotiation module with the resource, which cannot satisfy user's QoS demands. It finishes the execution of grid service and reminds the user.

**Grid Resource Node.** The Resource Information Provider Service module locates in Grid Resource Node, which is used for monitoring the QoS information of physical resources in grid. It obtains the newest information of resources through the QoS Control module, and provides the Resource Information Service Center module and the Resource Monitor module with renewed information. If the result of the QoS negotiation is that it is able to provide resource that can satisfy user's demand, the QoS Admission Control module would complete tasks such as resources co-allocation, conflict detect, deadlock detect and load balance, etc. Then, finally the module will finish the affirmation work requested by the user. The Trade Server module is responsible for determine the using price and record the information such as the total cost of the used resource and the user's account information etc. The Resource Reservation module is responsible for setting resources reservation flag and sending grid job to the Waiting-job Pool, waiting to be scheduled. Otherwise, Waiting-job Pool should responsible for adjusting the priority of grid jobs dynamically. The Scheduler takes charge of the scheduling of jobs in Waiting-job Pool according to particular strategy. In general, the priority of local job is higher than that of the grid job. It is permitted that the grid job has higher priority when grid job is very close to its deadline. The QoS Control module takes charge of the control of all dynamic QoS parameters. It adjusts QoS parameters according to the result of QoS negotiation, such as bandwidth, buffer size, etc. It should also response the inquiry from the Resource Information Provider Service module and renews its state information.

## 4 2-Phase Trust-Based Scheduling

### 4.1 Definition of Trust

Grid system is a distributed, dynamic adjustment-enabled computing platform, so the trust [8] [9] among the nodes of grid system should also meet the requirements to be distributed and adjusted dynamically. Furthermore, considering the model of human society, the trust between each node should also include the following two parts:

- **Direct Trust:** The trust brought by direct interaction and cooperation between source node and destination node, obtained directly from the source node;
- **Reputation:** The trust brought by direct interaction and cooperation between non-source node and destination node, which will be recommended to the source node by other non-source nodes.

In WAN, since direct cooperation often exists between node  $i$  and node  $j$  in one certain progress of computing task, the definition of direct trust from source node  $i$  to target node  $j$  can be shown as follows:

$$Trust_{(i,j)}^D = \left( \sum_{k=1}^N \alpha_k p_k^j \right) \times (1 - \rho^j) \quad (1)$$

Here,  $p_k^j$  is the statistic parameter of node  $j$ ,  $N$  is the total number of parameters,  $\alpha_k$  is the weight,  $\rho^j$  is the systematic crash rate of target node  $j$ .

The evaluation of statistic parameter  $p_k^j$  is very important, and it's also very difficult to ensure that the evaluation is both accurate and rational. For example, we can set  $R_l^e$  to be estimated response time for the  $l$ -th computing, set  $R_l^a$  to be actual response time for the  $l$ -th computing, then the measure of response time for the  $l$ -th computing,  $ResTime_l^j$ , is :

$$ResTime_l^j = \begin{cases} 0, & \text{when : } R_l^a = R_l^e \\ 0, & \text{when : } R_l^a = R_l^e \\ 1, & \text{when : } R_l^a < R_l^e \end{cases}$$

$p_l^j$  is the statistic value of every measure of response time  $ResTime_l^j$ , that is  $p_l^j = \sum_l ResTime_l^j$ . Objectively view, the more  $p_l^j$  is, the better the performances of node  $j$  on the measure of response time are, and this will also illuminate that node  $j$  has much more powerful computing ability. But unfortunately, after careful analysis, it's shown to us that the evaluation of estimated response time,  $R_l^e$ , will do direct effect on the measure of response time  $ResTime_l^j$ , and further more, on the evaluation of  $p_l^j$ . Based on it, the estimated response time  $R_l^e$  for every computing process may need rational and accurate evaluation.

Reputation is a kind of trust between target node  $j$  and no-source nodes via direct cooperation. It's always recommended to source node by other no-source nodes. The definition of reputation can be:

$$Reputation_j = \sum_{\substack{k=1 \\ k \neq i, j}}^M (\beta_k \times Trust_{(k, j)}^D) \quad (2)$$

Here,  $M$  = total number of no-source nodes  $-1$ ,  $\beta_k$  is the measure of relationship between source node  $i$  and no-source node  $k$ , such as the successful cooperation-rate. We must point out that the measure of relationship can be dynamically regulable, and  $\sum \beta_k = 1$ ,  $\beta_k = 1/M$  initially.

Since trust between source node  $i$  and target node  $j$  consists their direct trust,  $Trust_{(i, j)}^D$ , and the reputation of target node  $j$ ,  $Reputation_j$ , the definition of trust between source node  $i$  and target node  $j$ ,  $Trust_{(i, j)}$ , can be shown as follows:

$$Trust_{(i, j)} = \alpha \times Trust_{(i, j)}^D + \beta \times Reputation_j \quad (3)$$

Here,  $\alpha$  and  $\beta$  are the weights of direct trust,  $Trust_{(i, j)}^D$ , and reputation,  $Reputation_j$ , respectively. By setting these two parameters, the key factors of the trust between source node  $i$  and target node  $j$  can be shown obviously.

## 4.2 2-Phase Scheduling

Since grid computing system is an Internet-based, distributed computing platform, it involves not only LANs, but hosts in every LAN also. According to so, when a

scheduling is put to a computing task, it must be considered to schedule not only a certain LAN with a certain algorithm, but also a certain host in LAN with other algorithms. So a certain scheduling process can be divided into two levels:

- **External Scheduling:** WAN-scope, the first sub-task level, distributed scheduling;
- **Internal Scheduling:** LAN-scope, the second sub-task level, concentrated scheduling.

Notice that, sub-task partition is the most important key technology in grid computing system. We always hope that one computing task can be partitioned into sub-tasks hierarchically, and every sub-task also can be partitioned linearly. But unfortunately, it's only an idealistic view and it's always very hard to us to do so. The inner main reason is the extreme couplings and relationships between every tasks or sub-tasks. Because researches on sub-task partition is not the emphasis of the paper, and to simplified our research model, the paper assumes as premises that one computing task can be partitioned into sub-tasks hierarchically, and every sub-task also can be partitioned linearly.

External scheduling algorithm works in WAN-slope, and it schedules the first level subtasks. By using formula (4), it evaluates the estimated response time  $R_j^e$ , which computing a first level subtask in a certain LAN  $j$  as a node is needed.

$$R_j^e = E_j^e + T_j^e + Sch^e \tag{4}$$

Here,  $E_j^e$  is the estimated execution time which computing this first level subtask in LAN  $j$  is needed, and

$$E_j^e = \frac{\text{Total workload of this first level subtask}}{\text{Computing ability of LAN}_j} \tag{4.1}$$

$T_j^e$  is network transmission time cost of this first level subtask between source  $LAN_i$  and target  $LAN_j$ , and

$$T_j^e = \frac{(\text{Description of this first level subtask} + \text{Description of result of this first level subtask})}{\text{Measure of network transmission between } LAN_i \text{ and } LAN_j} \tag{4.2}$$

$Sch^e$  is the estimated scheduling time cost, and

$$Sch^e = \text{Complexity of algorithm} \times \text{Complexity of Network} \tag{4.3}$$

Not importing the mechanism of trust, external scheduler evaluates relevant estimated response time  $R_j^e$  at first, then sorts every  $R_j^e$  in decline order, locates the LAN with smallest  $R_j^e$ , and finally schedules a certain the first level subtask to this LAN.

Internal scheduling algorithm works in LANN-slope, and it schedules the second level subtasks. By using formula (5), it evaluates the estimated response time  $R_j^e$ , which computing a second level subtask on a certain host  $j$  as a node is needed.

$$R_j^e = E_j^e + T_j^e + Q_j^e \tag{5}$$



Here,  $E_j^e$  is the estimated execution time which computing this second level subtask in host  $j$  is needed, and

$$E_j^e = \text{Total workload of this second level subtask} / \text{computing ability of } HOST_j \quad (5.1)$$

$T_j^e$  is network transmission time cost of this second level subtask between source  $HOST_i$  and target  $HOST_j$ , and

$$T_j^e = (\text{Description of this second level subtask} + \text{Description of result of this second level subtask}) / \text{Measure of network transmission between } HOST_i \text{ and } HOST_j \quad (5.2)$$

$Q_j^e$  is estimated queuing time cost on one host  $j$ .

Without importing the mechanism of trust, internal scheduler evaluates relevant estimated response time  $R_j^e$  at first, then sorts every  $R_j^e$  in decline order, locates the host with smallest  $R_j^e$ , and finally schedules a certain second-level subtask to this host.

### 4.3 2-Phase Trust-Based Scheduling

Although the concept of Trust imbibes the merit of descriptions of relativities among human beings in our society, the grid computing system is a very complex and hierarchical architecture, for it consists of LANs and hosts. 2-Phase scheduling algorithm is the right one to show out this hierarchy explicitly.

Based on the reasons above, we import the mechanism of trust into 2-Phase scheduling, and then present our 2-Phase Trust-Based Scheduling Algorithm (2PTBSA). Our target is to avoid unstable nodes during computing progress, and to enhance the total computing efficiency, by the filtration with the mechanism of trust, on the premise of better descriptions of the complexity and hierarchy. The proposed algorithm can be described as follows:

- (1) According to 2-Phase Scheduling of Tasks, evaluate the estimated response time  $R_j^e$  for every LAN  $j$  in WAN-slope;
- (2) Importing the mechanism of trust, evaluate candidate  $W_j$  by using formula (6), here LAN  $i$  is the right LAN the target node is located;  $W_j = \frac{1}{Trust_{(i,j)}} \times R_j^e \quad (6)$
- (3) Sorts  $W_j$  in decline order, locate the LAN  $j$  with the smallest  $W_j$ , schedule a first level subtask to compute or to deeper (the second level) scheduling;
- (4) According to Internal Scheduling, schedule the second level tasks in every LAN  $j$  located (considering relatively reliabilities of hosts and resources in LAN);
- (5) Run a statistic operation on the actual response time  $R_j^a$  in LAN  $j$  located, update  $Trust_{(i,j)}^D$  between LAN  $i$  source node located in and LAN  $j$  target node located in;
- (6) Update Reputation <sub>$k$</sub>  of other LAN  $k$ ;
- (7) Update  $Trust_{(i,j)}$  between LAN  $i$  source node located in and LAN  $j$  target node located in;
- (8) Redraw resources which computing used, clear all buffers which computing sessions used.

## 5 Conclusions and Future Work

To get higher efficiency of resource allocation and management and task scheduling algorithm in grid computing, this paper, based on the analysis of the related work on grid resource management and task scheduling, presents a QoS based structure for Grid Resource Allocation and Management System (GRAM) and a 2-Phase Trust Based Scheduling Algorithm (2PTBSA).

After analyzing the 2PTBSA algorithm in detail, it can be concluded that: (1) 2PTBSA algorithm can reduce the actual response time of one certain computing task. The reduction of the actual response time mainly lies on 2PTBSA algorithm, which can avoid unstable (or distrusted) LANs and hosts as much as possible during its scheduling. (2) Even if it cannot avoid unstable LANs or hosts, 2PTBSA algorithm will try its best to avoid the LANs and hosts which crash frequently. The analysis shows that the damages caused by few, iterative and concentrated crashes are much more serious than the ones caused by a little more, distributed and nonrecurring crash. (3) The effect of 2PTBSA algorithm equals the one of 2-Phase algorithm under the condition of equipotent crash rates and times.

In the field of grid resource management, through analyzing the QoS characteristics in grid, we propose and set up the layered structure of Grid QoS, which provides a reasonable gist for mapping and converting QoS parameters in grid, On the basis of analyzing the content of GRAM based on QoS, we put forward the architecture of QoS-based GRAM. It provides a reasonable model for the QoS and resource allocation management in grid. However, there still remains some work to be improved. Because there are many uncertain factors for grid QoS and the relationships among nodes are more complicated in grid environment than in traditional networks, how to map, analyze and convert the users' QoS demands to concrete QoS parameters effectively will be our next research topic. We plan to design a simulation platform supporting Grid QoS to test the usability of GRAM-QoS and enhance its performance. We also aim to implement GRAM-QoS on the Globus toolkit.

Meanwhile, although the 2PTBSA Algorithm can describe the trust level of those parameters which can enable the task scheduling to evade unstable nodes and improve the efficiency of the whole computational task, those influential parameters in task scheduling need to be observed in the extensive network environment for a long time. Also, the study on task scheduling about these parameters is still at the simulation and emulation stage at present, that is to say we can only simulate the emergence of these factors by using some theories such as queuing theory, probability theory, and then implement corresponding task scheduling algorithm at these unstable points. Considering this, a grid simulation platform, which can estimate the metrics in different fields, will be another key point of our future work.

## Acknowledgements

This work is supported by National Natural Science Foundation of China under Grants No. 90204009 and 90412014, by China Specialized Research Fund for the Doctoral Program of Higher Education under Grant No. 20030286014, and by Jiangsu

Provincial Key Laboratory of Network and Information Security under Grant No. BM2003201.

## References

1. Foster, I.: What is the Grid? A three point checklist. Argonne National Lab and University of Chicago, (2002) <http://www-fp.mcs.anl.gov/~foster/Articles/WhatIsTheGrid.pdf>
2. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. <http://www.globus.org/research/papers/ogsa.pdf>, Globus Project (2002)
3. Chen, H., Maheswaran, M.: Distributed dynamic scheduling of composite tasks on grid computing systems. In Proceedings of 16th International Parallel and Distributed Processing Symposium (IPDPS 2002), (2002) 88-97
4. Min Rui, Maheswaran, M.: Scheduling co-reservations with priorities in grid computing systems. In Proceedings of 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID2002), (2002) 250-251
5. Subramoniam, K., Maheswaran, M., Toulouse, M.: Towards a Micro-Economic Model for Resource Allocation in Grid Computing Systems. In Proceedings of 2002 IEEE Canadian Conference on Electrical & Computer Engineering (2002) 782-785
6. Al-Ali, R.J., ShaikhAli, A., Rana, O.F., Walker, D.W.: Supporting QoS-based Discovery In Service-oriented Grids. In Proceedings of 17th International Parallel and Distributed Processing Symposium (IPDPS 2003), (2003)
7. Buyya, R., Abramson, D., Giddy, J.: A Case for Economy Grid Architecture for Service Oriented Grid Computing. In Proceedings of 15th International Parallel and Distributed Processing Symposium (IPDPS 2001), (2001) 776-790
8. Azzedin, F., Maheswaran, M.: Evolving and Managing Trust in Grid Computing System. In Proceedings of 2002 IEEE Canadian Conference on Electrical & Computer Engineering (2002) 1424-1429
9. Azzedin, F., Maheswaran, M.: Towards Trust-Aware Resource Management in Grid Computing Systems. In Proceedings of 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID2002), (2002) 419-424

# Learning to Plan the Collaborative Design Process

Flávia Maria Santoro<sup>1,2</sup>, Marcos R.S. Borges<sup>2,\*</sup>, and Neide Santos<sup>3</sup>

<sup>1</sup> Federal University of State of Rio de Janeiro – UNIRIO,  
Av. Pasteur 458, CEP 22290-040, Rio de Janeiro, Brazil  
flavia.santoro@uniriotec.br

<sup>2</sup> Graduate Program in Informatics – Federal University of Rio de Janeiro,  
PO Box 2324, CEP 20001-970, Rio de Janeiro, Brazil  
mborges@nce.ufrj.br

<sup>3</sup> State University of Rio de Janeiro UERJ, Rua São Francisco Xavier 524,  
CEP 20559-900, Rio de Janeiro, Brazil  
neide@ime.uerj.br

**Abstract.** A key factor in team collaboration is the process followed by participants. Planning this process helps teams to accomplish their goals. In collaborative design environments, process modeling should be used to make the collaboration explicit. In Computer-Supported Collaborative Learning (CSCL) research, we have observed that most environments do not provide support for the definition of collaborative processes by apprentices and facilitators. This support is very important when the approach is based on project-based learning (PBL). Planning the interactions and the process of a project is a way to stimulate group participants to collaborate, thereby promoting interdependency and commitment between the work and the learning activities. We present a cooperative project-based learning environment that supports facilitators and apprentices in the task of planning their work in order to improve awareness about the “learning-how-to-learn” process.

## 1 Introduction

The demand for sophisticated and customizable products, together with competitive pressures to reduce costs, will increase considerably over the next decade. Due to growing global competition, an increasing number of products has been developed through intensive collaboration among people and organizations. As a result, collaborative product development (including design, manufacturing, operations, and management) has become a new paradigm for today’s engineering organizations. Team effort from all project participants (working in harmony with the project goals, timetable, and budget) is critical for product design. The need for timely, reliable communication is an essential ingredient for team work, which now likely spans engineering and manufacturing workgroups that are distributed throughout the enterprise and supply chain worldwide [2] [14].

Collaborative product development often starts with a complicated process that involves groups of designers, manufacturers, suppliers, and customer representatives,

---

\* On sabbatical leave at DSIC-Polytechnic University of Valencia, Spain.

each of which has their own objectives. Because of the involvement of many stakeholders in decision-making, numerous design conflicts arise at every stage of a collaborative engineering design process. Thus, an effective Collaborative Virtual Environment (CVE) is required. Collaboration, however, is not as natural and automatic as one might think. It needs to be planned and supported even for task-oriented group activities. The key aspect behind effective collaboration is the process.

The design of the process involves process planning as the major activity [21]. Process planning demands collaborative work, information sharing, and the exchange of ideas among multiple parties in different locations. All this requires the skill and experience that are acquired when teams work together over time or through specific training.

CSCL (Computer-Supported Collaborative Learning) is a research area that studies how groups learn while they interact to perform their educational tasks. Many CSCL proposals present systems that provide a project-based learning (PBL) approach. In the analysis of these systems, we have observed that many support the execution of specific tasks in the context of a project; an environment that is very similar to that found in collaborative design. Planning the interactions and the process in a collaborative project is a way to stimulate people to collaborate, while promoting interdependency and commitment to their work.

It can generally be observed that virtual training environments do not provide support for the definition of collaborative processes, or any support for all the stages of a development project, even though many researchers have stressed the importance of this feature. One of the main goals of a collaborative learning environment is to train people to design their collaboration process. This will help them not only to plan their activities in a collaborative environment, but, most importantly, to teach people how to collaborate.

This paper proposes a training infrastructure based on a cooperative project-based environment where the main goal is to support team members in the task of planning their work in order to improve awareness about the design process. Collaborative planning is not an easy task and people should be trained for this in the context of Collaborative Virtual Environments [18].

The paper is divided as follows. Section 2 presents an overview of the requirements for a collaborative design environment. In Section 3, we discuss the characteristics of CSCL systems and the importance of planning in the Process Based Learning approach. Section 4 describes our proposal for providing support to collaborative processes. Section 5 presents the results of the experimental work we conducted. Section 6 presents our conclusions and future work.

## **2 Collaborative Design**

Today many types of organizations, including industry, need to manage their product development life cycle and implement change management through collaborative engineering. Product design takes place in a collaborative environment, through real-time information exchange among engineering and manufacturing teams, suppliers, customers and partners [21]. Collaborative engineering and project management can

be carried out in collaborative virtual environments where the teams exchange data such as project plans, documents, and product structures.

Collaborative virtual environments have been used for product and system design, and manufacturing activities, providing simulation, viewing, reviewing, and iteration of parts, etc. Collaborative virtual tools give program designers, engineers, developers, maintainers, trainers, and even eventual users with an expanded capability to concurrently make manufacturing design more productive and efficient.

Collaborative environments facilitate not only engineering excellence, but also purchasing, manufacturing, and logistic efficiencies throughout the supply chain. By streamlining the flow of information between each member of the project, Collaborative Design helps organizations drive innovation at a faster pace. Questions or issues about the optimal design of a product can be quickly identified and resolved.

A large number of papers on distributed collaborative design and Internet-based education have been published. Numerous commercial products are also available on the market, which implement partial features of distributed collaborative concepts [23]. Sun and Gramoll studied the area and observed that research on distributed collaborative design falls into two categories: the first offers theoretical models, and the second implements the idea of distributed collaborative design in a practical way. The authors describe and compare the most important applications found in the literature [23].

Pahng, Senin and Wallace proposed a distributed object-based modeling and evaluation framework (DOME) for product design [15]. The goals of DOME are to link distributed design modules, to aid designers in evaluating the system performance with different design alternatives, to seek optimal solutions, and to make design decisions. Case and Lu proposed a discourse model used in software environments that provides automation support for collaborative engineering design [4]. The model treats interactions between designers as a discourse process.

Lee, Kim, and Han developed a prototype to implement web-enabled feature-based modeling in a distributed environment [12]. Cybercut is a web-based design system for manufacturing, which was developed at the University of California, Berkeley [22]. It provides Internet-based services such as design-for-manufacturing CAD, Computer Aided Process Planning (CAPP), and access to an open architecture machine tool for manufacturing of mechanical parts. Mori and Cutkosky proposed an agent-based prototype implementation in the design of a portable CD player [13]. To seamlessly share CAD information throughout an enterprise, major CAD suppliers have introduced a software system called Product Development Management [8], [16], [17]. This software system extends CAD data not only to non-design departments of companies such as analysis, tooling development, manufacturing, testing, quality control, sales and marketing, but also to the suppliers and partners of these companies.

Sun and Gramoll [23] assert that in addition to conducting the design of real products over the Internet, virtual environments can also support engineering education and organizational training.

One of the most important facilitators of the collaborative aspects of manufacturing and design is the virtualization of the process components. This virtualization requires support technologies and, most importantly, a real collaborative environment. Collaboration, however, requires more than support technologies, which are currently

provided by groupware. Collaboration requires both attitude and team awareness from the participants.

A collaborative attitude is a cultural matter. It needs to become part of participants' way of thinking in order to produce a collaborative environment. Only extensive training and drilling can achieve this. On the other hand, the participants' collaborative attitude also requires support from the system. Participants need to be aware of each other's activities. When activities are made explicit to all participants, they become aware of their interdependencies and the need for collaboration. This is required not only for the planning stage, but also during the process enactment.

### **3 Planning the Process in Project-Based Learning**

The pedagogical work with projects portrays a posture through which the facilitator organizes learning situations based on the apprentices' spontaneous and significant discoveries. This induces the apprentices/authors to think about their actions and become capable of developing and creating a product that reflects their learning process, synthesizing the knowledge that is built [5], [10]. The foundation for the method is situational learning [3], [11]. In this method, the apprentice builds knowledge, attributing his/her own meaning to the contents, and the transformation of data that comes from different sources is understood from common sense.

In a project, learning occurs by interaction and articulation among different knowledge areas in order to foster the construction of autonomy and self-discipline. It also aims to develop abilities for working in teams; abilities such as decision making, facilitation of communication, problem formulation and problem solving. A project can be described as a process that is divided into stages that are related to each other, forming a flow of work. Each stage is made of the performance of one or more activities. It has specific objectives and generates some kind of product. These activities should stimulate information sharing and knowledge building [6].

During the flow of work, the following goals can be achieved: (i) maintenance for the collaborative posture, awareness of each other's responsibility for work in the group; (ii) an understanding of each stage's goals in the global context; (iii) motivation for the maximum interaction among the participants. At the operational level, a project's flow of work consists in the preparation, implementation, and posterior performance of educational activities by apprentices interacting in groups. Thus, the clear definition of the activities in a project is essential to establish the positive interdependence required to stimulate the collaboration. In a collaborative learning environment, the facilitator and the apprentices should have the means to define educational processes, to configure different scenarios and projects, and to obtain support, through the available tools, in the accomplishment of their tasks.

For George and Leroux [7], a project should be structured over time and divided into successive stages, forming an action plan. The careful planning of the activities is necessary to provide the project with a temporary structure and the description of human activities as the actions performed through operations help the understanding of the fundamental role that the planning has in human cognition. The planning based on previous experiences advances the results of the actions; nevertheless, these

anticipations should be implemented and adjusted according to the real situational conditions [9], [24].

A workflow model can be useful to represent the flow of activities. A typical workflow system helps to define, execute, coordinate and monitor the business processes in an organization. Therefore, the system should contain a representation of the activity structure and the work procedures. This representation is usually a sequential or hierarchical decomposition of an activity into tasks, which is built separately from the execution of the activity. Van der Veen et al. [25] carried out experimental studies and concluded that the use of workflow systems as a support to project-based learning enhances the educational goals.

In contrast to this classic model, Bardram [1] states that instead of supporting the information routing, the process in workflow systems should mediate the reflection and anticipation of appealing events in the work. Thus, a planning tool should support the construction, change, execution, and monitoring of the collaborative activities throughout their development.

In our approach, we have used the concepts of a collaborative process in the same way as they are defined in workflow systems. However, we have focused mainly on the planning of the project, the definition of the interactions among the group members, the definition of responsibilities, and the strategies for the solution of the problems. In the case of educational processes, there is a commitment to the learning of assimilated concepts. Therefore, it is necessary to count on the facilitators' experience to plan the projects and to think of situations that stimulate the apprentices to work in a collaborative way. It is also necessary to lead the apprentices to consider hypotheses of solutions for problems defined in the projects, to discuss and ponder them, and to generate final products being aware of the process performed by them in order to achieve their goal.

## 4 COPLE: Cooperative Project-Based Learning Environment

The CSCL infrastructure called COPLE – *Cooperative Project-Based Learning Environment* – has implemented the proposed ideas about the design of learning processes [19]. The core of COPLE is a process server that is responsible for the execution of the process and works as a workflow machine. An editing tool is necessary to design the collaboration processes (Figure 1). The definition of the activities and their execution flow allows for the design of the interaction between the participants of the process and the products generated during the project development. The design allows project leaders to have a good understanding of the processes and also to monitor the executors' work.

COPLE was supplied with a Cooperative Process Editor (COPE) to design the collaboration processes. COPE adopts standard symbols and conventions to represent the process components: activities, roles, agents, flow, rules, descriptions, and the relationships among them. The process server interprets the model, starts the process execution at the beginning of the project, and maintains the relevant information about the process activities. The main interface of COPE (Figure 2) presents the shared space for the graphical edition of the process.



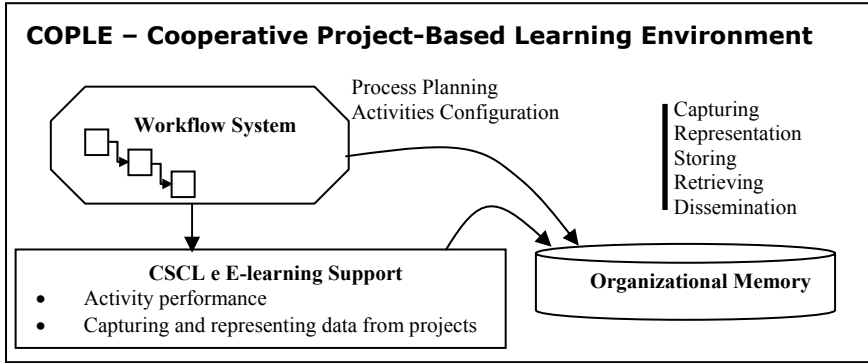


Fig. 1. The Representation of the COPLE Architecture

A process is a flow of activities. The participants should describe each activity in detail in the COPLE groupware tool as shown in Figure 3. Each activity should be configured by defining its attributes and characteristics. A general description of the activity as well as its duration and type should also be configured.

COPE provides the definition of rules, resources, and role assignment. Roles are selected from a pre-defined list. These roles associate members with specific tasks. Tasks can be paired with a supporting tool and a document that will be handled by the tool.

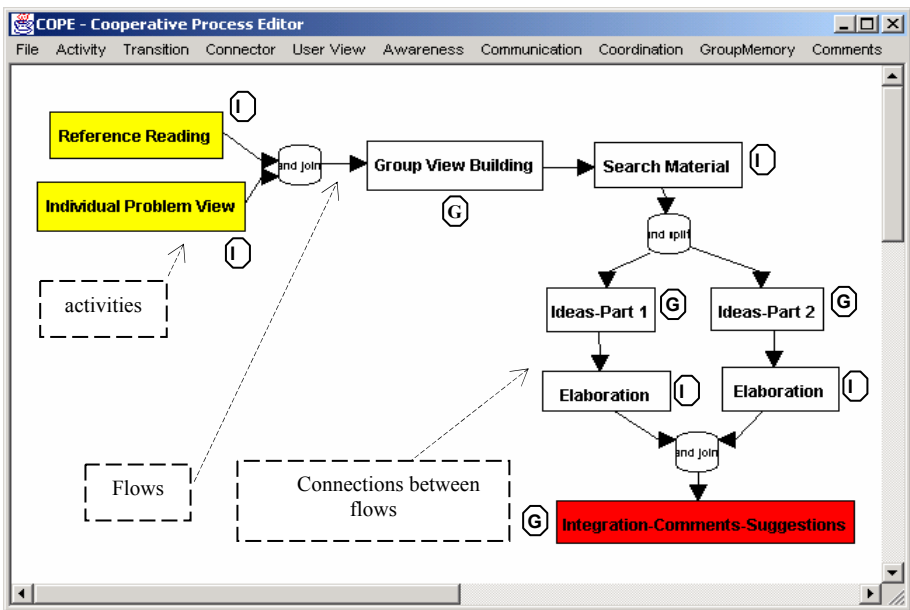


Fig. 2. An Example of Educational Process Design using COPE

The screenshot shows the 'Activity Edit' dialog box with the following configuration:

- Activity Name:** Group View Building
- Type:** Task
- Type Interaction:** GROUP
- Description:** The group should build a collective view of the problem based on the individual ideas and the material read in the previous activities.
- Duration:** 2 days
- Rules:** The coordinator gathers all the material generated. Each member should read everything. Each member argues and proposes a consensus.
- Roles:** Coordinator, Writer, Editor, Reviewer
- Resources:** Texts sent by each group member.
- Supported by Tool:** Text Editor
- Document Name:** GroupView

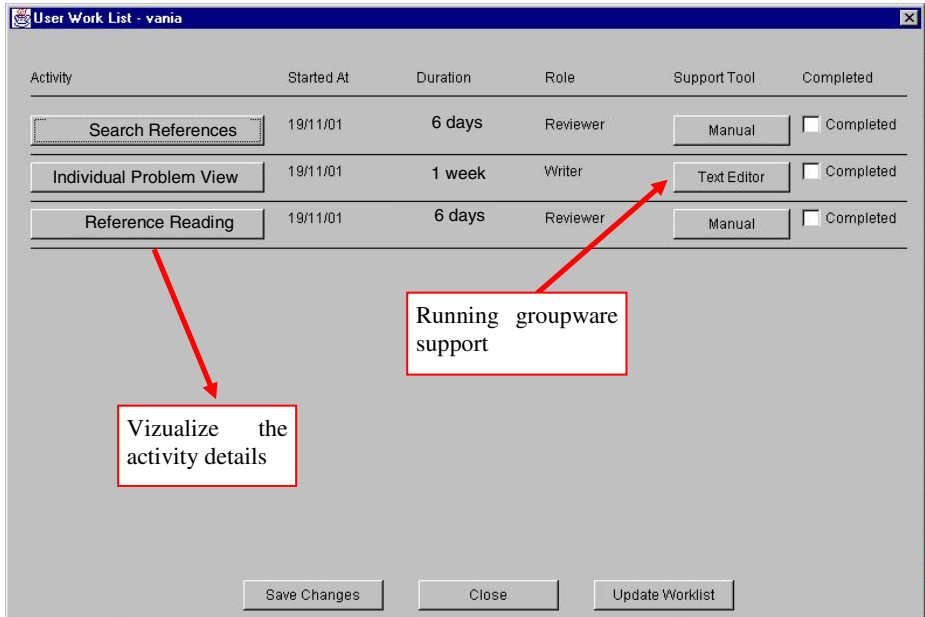
**Fig. 3.** Activity Configuration in COPE

After discussing all the issues related to the process, the group members agree on a single process definition. This definition represents a contract that establishes how they will work together. The explicit representation and the understanding of individual responsibilities help participants learn how to collaborate. They learn the topics and concepts of the subjects being taught as well as the issues that are related to project development and interdependency.

The process describes the strategy of the group's interaction for reaching the final product taking into account everyone's contribution and participation. The importance of the transition from individual tasks (I) to group tasks (G) is highlighted, making the performance of each group member clear. At a certain point, the work is divided into parts. However, each part requires total group involvement because the entire group generates ideas for further elaboration. Processes used in previous projects can be a good source of learning for the groups. They should be considered as part of the group memory and can be reused, refined, and adopted by other groups.

After the group members and facilitators define a process, they can start working. They access their individual Work Lists (Figure 4), where they can find the information that is relevant to the tasks they need to perform. They can also access the supporting tools to help complete their tasks. The items that appear in these Work Lists are the activities assigned to the participant at that particular time. When the participant completes the activity and informs the system, the process server executes the next process step and then presents the next tasks to be carried out to group members. This is repeated until the process has been completed.

Specific groupware tools, such as the Cooperative Text Editor [20], support some activities in a project. Apprentices use these tools to build products, discuss related



**Fig. 4.** Access to the Activities through the Work List

issues and save their findings, as they work together to reach their goals. The group also has a message and chat system to communicate and interact within the integrated environment. The group interaction is directly associated with each activity and its products.

Collaborative tools can be evoked from the Work List Interface, which starts a specific work session automatically, in order to manipulate the product (document) that is related to the activity. All the tools are integrated in the COPLE environment; therefore, the activities and interactions performed can be traced. The entire group also shares all documents, which helps to build and evaluate the project memory.

## 5 Results from Case Studies

Four case studies were carried out in regular classes at three different universities in order to perform a preliminary evaluation of our approach. Our main assumption was that by fostering discussion and defining the working processes, the collaboration among group members would improve. In all four cases, we proposed that the groups develop projects that deal with the design of a solution to a stated problem, the discussion of questions related to it, and the writing of a report about their findings and their conclusions. We had some groups working under an explicit defined process and others working in an ad-hoc manner.

We attempted to evaluate the level of collaboration among the group members during the project development using four aspects: communication, collective knowledge building, coordination and awareness. The analysis was mainly qualitative.

Communication is related to the quality of messages exchanged among the participants. Collective knowledge building refers to the process of sharing ideas and building an artifact together. It was evaluated by observing the participants' contributions to discussions and in the resulting document. We also considered the comments and the suggestions from other members as contributions. Coordination refers to the ways in which groups articulate and design their strategies to solve a problem. Awareness is the perception that a member has of the other members' activities, their participation and the way the entire group carried out the work.

**Table 1.** Summary of Results

<b>1<sup>st</sup> Case Study</b>	<p>Six groups of four people geographically distributed: three groups had explicit processes and the others did not. They were asked to choose the coordinator. The facilitator defined the work processes and presented it to the three groups. The groups worked without much support from the facilitator.</p> <ul style="list-style-type: none"> <li>➤ Groups with well-defined processes present better results, regarding the amount of interaction and the contribution to the final product.</li> <li>➤ The texts that were produced by the groups reflected the number of contributions, but it was not possible to identify the relationship among them.</li> </ul>
<b>2<sup>nd</sup> Case Study</b>	<p>One group had to define their work process alone and recall all information about the development of their tasks. They did it alone, without the facilitator's intervention.</p> <ul style="list-style-type: none"> <li>➤ The group reported the feeling that defining a work process and following their initial planning had contributed to improving peer collaboration.</li> <li>➤ In the questionnaires answered by the participants, there was evidence of the occurrence of individual contributions through discussions and interactions during the construction of the text.</li> </ul>
<b>3<sup>rd</sup> Case Study</b>	<p>Each of the two groups were asked to develop two similar projects. At first, they worked in an ad hoc manner. In a second occasion, they used COPLE to plan and perform their processes. The facilitators helped the apprentices to define their processes.</p> <ul style="list-style-type: none"> <li>➤ The results, particularly from Group 2, showed collaboration promoted the evolution from an ill-defined project to a well-defined project. The coordination helped the participants to establish better relationships with each other and greater commitment to their work.</li> <li>➤ The relationships among individual contributions were higher in the second project for both groups, where the groups had explicitly designed their working processes.</li> </ul>
<b>4<sup>th</sup> Case Study</b>	<p>Six groups of four people working together in a lab: three groups had explicit processes and the others did not. The facilitator helped the apprentices to define their processes and suggested forms and rules of interaction among the participants in the group.</p> <ul style="list-style-type: none"> <li>➤ The groups that used explicitly defined processes presented higher levels of collaboration than the groups that worked without following a process. The three groups with defined processes generated higher quality products. We believe that this was due to a higher number of contributions.</li> <li>➤ The groups that had planned their work presented better results in the co-building of the text, both in quality and in length.</li> </ul>

Interaction data were obtained from the environment and the participants filled out a questionnaire after the completion of the project. With these two instruments, we tried to assess the individual satisfaction and their perception of the collaboration level.

The composition of the groups, the nature of the task, the collaboration context, and the infrastructure for communication are key points for successful collaboration in groupware. In all four case studies, the tasks were similar; all of them involved the collective production of texts. However, the work processes for each group were quite different.

Although the groups consisted of people with similar academic skills, we observed great differences in their work dynamics. Individual characteristics have a strong influence on groups. Therefore, environments should stimulate and exploit the individuals' characteristics in order to improve the quality of the work. We also tested the presence/absence of the coordinator's/facilitator's support. A summary of the results is presented in Table 1.

In general, we observed that communication within groups with a detailed designed process presented a higher number of messages. This suggests that interaction was intense in these cases. Participants were conscious about their goals and roles in the project, and they also knew when, how, and with whom to interact in order to accomplish their tasks.

These observations suggest that the mechanisms for the definition and follow-up of processes really help to stimulate collaboration in work groups. However, interaction based on collaboration support technologies has not yet been widely used. In COPLE, the mechanisms for structuring and representing knowledge, such as a typology for chats and messages to aid communication, are still being implemented. If these mechanisms were present, they would improve the understanding of the tasks, the relationship between contributions, the communication between group members, and, consequently, the level of collaboration.

## 6 Conclusions and Future Work

The findings of the case studies have confirmed our approach and provided the basis for new research in the CSCL area. It is important to continue our research to be able to generalize the results achieved so far. Formative evaluation, such as the one used in our work, can bring valuable preliminary results. In spite of the size limitation of the case studies, many problems have been revealed and can be resolved in future experiments.

The general conclusion is promising as far as the confirmation of the hypothesis about the relationship between the process design and the level of collaboration. We have addressed the main issues and that we have succeeded in resolving some problems in project development and in stimulating collaboration using a simple workflow mechanism. We intend to continue using the COPLE environment to analyze its applications in other situations, by presenting new problems to apprentices and also using different facilitators.

The challenge for designers of collaborative learning systems is to create the computational support that allows apprentices to follow the necessary paths for the development of the proposed project. If this is attained, it would provide a theoretical and practical perspective for relating educational objectives and technical innovations. The study of interactive strategies among apprentices' groups and the integration of collectively built knowledge should also be included. The approach described in this paper is a first step towards combining these two dimensions.

## Acknowledgments

Marcos R. S. Borges was partially sponsored by a grant from the Secretaria de Estado de Educación y Universidades of the Spanish Government.

## References

1. Bardram, J.E.: Plans as Situated Action: An Activity Theory Approach to Workflow Systems. Proceedings of European Computer-Supported Collaborative Work Conference – ECSCW. Lancaster, UK (1997) 17-32
2. Bochenek, G. (moderator): Collaborative Virtual Environments to Support System Design and Manufacturing. Panel Session at the International Symposium on Collaborative Technologies and Systems Orlando, USA (2003). Available at: <http://www.engr.udayton.edu/faculty/wsmari/cts03/panel.htm>, accessed in April 2005.
3. Brown, J.S., Collins, A., Duguid, P.: Situated Cognition and the Culture of Learning. Educational Researcher, Vol. 18 No. 1. American Educational Research Association (1989) 32-41
4. Case, M.P., Lu, S.C.: Discourse Model for Collaborative Design. Computer-Aided Design, Vol. 28 No. 5. Elsevier Science Ltd (1996) 333-345
5. Dewey, J.: Democracy and Education. Free Press, New York, USA (1996)
6. Ge, X., Yamashiro, A., Lee, J.: Pre-class Planning to Scaffold Apprentices for Online Collaborative Learning Activities. Educational Technology & Society, Vol. 3, No.3. International Forum of Educational Technology & Society (2000) 159-168
7. George, S., Leroux, P.: Project-Based Learning as a Basis for a CSCL Environment: An Example in Educational Robotics. Proceedings of First European Computer-Supported Collaborative Learning Conference. Maastricht, Holland (2001) 269-276
8. Global Collaborative Engineering. Available at: <http://www.enovia.com/>, Accessed January 2004
9. Grégoire, R., Laferrière, T.: Project-Based Collaborative Learning with Network Computers - Facilitators Guide. Canada. Available at: <http://www.tact.fse.ulaval.ca/ang/html/projectg.html>, Accessed in February 2005
10. Kilpatrick, W. H.: Foundations of Method: Informal Talks on Teaching. Macmillan, New York, USA (1926)
11. Lave, J., Wenger, E.: Situated Learning: Legitimate Peripheral Participation. Cambridge University Press, New York, USA (1991)
12. Lee, J.Y., Kim, H., Han, S.: Web-enabled Feature-based Modeling in a Distributed Design Environment. Proceedings of the ASME Design Engineering Technical Conference. Las Vegas, USA (1999) 12-15
13. Mori, T., Cutkosky, M.R.: Agent-based Collaborative Design of Parts in Assembly. Proceedings of the DETC-ASME Design Engineering Technical Conferences. Atlanta, USA (1998) 1-8
14. National Science Foundation: Multi-disciplinary Workshop at the Interface of Cyber infrastructure, and Operations Research, with “Grand Challenges” in Enterprise-wide Applications in Design, Manufacturing and Services: Final Report. Washington D.C., USA (2004). Available at: <https://engineering.purdue.edu/PRECISE/CI-OR/index.html>, Accessed in April 2005.

15. Pahng, G.F., Senin, N., Wallace, D.: Modeling and Evaluation of Product Design Problems in a Distributed Design Environment. Proceedings of the ASME Design Engineering Technical Conference. Sacramento, USA (1997) 14-17
16. Product Development Company. Available at: <http://www.ptc.com>, Accessed in January 2005
17. Product Lifecycle Management Solutions, <http://www.ugs.com/index.shtml>, Accessed in June 2004
18. Santoro, F.M., Borges, M.R.S., Santos, N.: Planning the collaboration process: one way to make it happen. Proceedings of the 8th International Conference on Computer Supported Cooperative Work in Design Vol II. Xiamen, China, IEEE Press (2004) 611-615
19. Santoro, F.M., Borges, M.R.S., Santos, N.: Learning through Collaborative Projects: The Architecture of an Environment. International Journal of Computer Applications in Technology – IJCAT: Special Issue on Computer-Supported Collaborative Work in Design, Vol. 16 No. 2/3. Inderscience (2003) 127-141
20. Santoro, F.M., Borges, M.R.S., Santos, N.: Experimental Findings with Cooperative Writing within a Project-Based Scenario. In: Computers and Education: Towards a Lifelong Learning Society. Kluwer Academic Publishers, London, UK (2003) 179-190
21. Shridhar, J.M., Ravi, S.: Virtual Manufacturing: an Important Aspect of Collaborative Product Commerce. Journal of Advanced Manufacturing Systems, Vol.1 No. 1. Society of Manufacturing Engineers (2002) 113-119
22. Smith, C.S., Wright, P.K.: Cybercut: A Networked Manufacturing Service. Journal of Manufacturing Systems, Vol. 15, No.6. Elsevier Science Ltd (1996) 1-11
23. Sun, Q., Gramoll, K.: Internet-based Distributed Collaborative Engineering Analysis. Concurrent Engineering, Vol. 10, No. 4. Sage Publications (2002) 341-348
24. Tiessen, E.L., Ward, D.R.: Developing a Technology of Use for Collaborative Project-Based Learning. Proceedings of Computer Support for Collaborative Learning. Stanford, USA (1999) 631-639
25. Van der Veen, J., Jones, V., Collis, B.: Workflow applied to Projects in Higher Education. Proceedings of Third IFIP Conference on Collaborative Information Systems WG 6.1 ECASP (Educational CAse Studies in Protocol) Workshop.). Paris, France (1998) 147-158

# Groupware System Design and the Context Concept

Marcos R.S. Borges<sup>1,\*</sup>, Patrick Brézillon<sup>2</sup>, Jose Alberto Pino<sup>3</sup>,  
and J.-Ch. Pomerol<sup>2</sup>

<sup>1</sup>NCE&IM, Universidade Federal do Rio de Janeiro, Brazil  
mborges@nce.ufrj.br

<sup>2</sup>LIP6, Université Pierre et Marie Curie, France  
{Jean-Charles.Pomerol, Patrick.Brezillon}@lip6.fr

<sup>3</sup>DCC, Universidad de Chile, Chile  
jpino@dcc.uchile.cl

**Abstract.** The concept of context can be advantageously applied to the Computer-Supported Cooperative Work field. The term *awareness* has traditionally been used in this area without explicit association to context. This paper attempts to clarify the relationship between these two concepts. In particular, a framework is proposed to understand context and awareness as connected to other concepts used in group work as well. The framework is useful to consider some groupware systems from the perspective of context and to obtain some insight on possible improvements for users. Two examples illustrate the application of the framework.

## 1 Introduction

The concept of context has not been well understood in the Computer-Supported Cooperative Work (CSCW) field. Context has been used in several publications in the area, but with several different meanings associated to it [8]. CSCWD (Computer-Supported Cooperative Work in Design) is a good example where context plays a role in the specialization of an area. Specialization, in this case, means the knowledge related to applying CSCW techniques in the area of Design. Nevertheless, contextualization seems so natural that people often lose sight of its real significance.

The meaning of the context concept depends on the subject area [6], [13]. On the one hand, there have been several conferences on modeling and the use of context since 1997 [7]. These events deal with aspects of context at the highest level of knowledge and reasoning. However, this approach rarely takes the practical aspects of context in real-world applications, such as collaborative work, into consideration. On the other hand, in CSCW articles, several issues point to context without referring to it as such. Context has been applied in group work and is usually associated with awareness mechanisms. Few groupware systems use the context concept to guide design decisions, leaving it to be processed mostly by users. Most misunderstandings are caused by not explicitly recognizing and representing the notion of context and its association with other elements of groupware systems.

We present a framework for understanding the concept of context in group work, and we also discuss the application of context in the area of CSCW. Our aim is to

---

\* On sabbatical leave at DSIC-Polytechnic University of Valencia, Spain.



guide the designer to the systematic use of context when developing an application [3]. We believe this model can be useful not only to understand the use of contextual information but also to relate components of groupware systems.

This paper is structured as follows: Section 2 reviews the concept of context; Section 3 presents a framework for understanding how groupware issues relate to context; Section 4 presents the groupware model for awareness mechanisms [3]; Section 5 uses the model to show cases where groupware fails in dealing with this concept; and Section 6 presents our conclusions.

## 2 Context

Context in real life is a complex description of knowledge about physical, social, historical, or other circumstances within which an action or an event occurs. Access to relevant contextual information is required in order to understand many actions or events. Understanding the “opening a window” action, e.g., depends on whether a real window, or a window on a graphical user interface is referred to [17]. It is possible (i) to identify various context types, and (ii) to organize them in a two-dimensional representation: vertically (i.e., depth first), from more general to more specific; and horizontally (i.e., width first), as a heterogeneous set of contexts at each level [5].

In the vertical representation (“depth first”), there are different contexts defined by their level of generality, mainly in highly organized systems. For example, the context of a building is more general (a higher level) than the context of an office. Contexts at a higher level contain general information, while contexts at a lower level contain more specific information. A context is like a system of rules (constraints) to identify triggering events and to guide behaviors in lower contexts. Based on Brézillon [4], it can be observed that a context at a general level contains contextual knowledge. The application of rules at more specific levels develops proceduralized contexts. A higher context is like a frame of reference for the contexts below it.

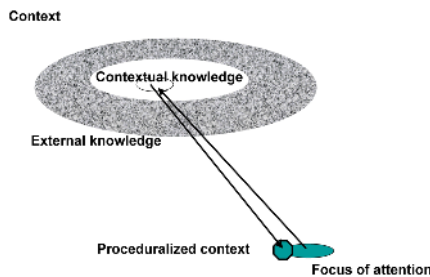


Fig. 1. Contextual knowledge and proceduralized context [6]

Each actor has its context in the horizontal representation (“width first”). The user’s context contains specific information; for example, the results of a meeting with a customer, the reasons for changing offices, etc. The context of a communicating object contains knowledge about its location, and how to behave with the other

communicating objects. Thus, at a given level of the context hierarchy, there is a set of heterogeneous contexts.

Pomerol and Brézillon [16] distinguish between the non-relevant and the relevant parts of the context for each step of a task. The non-relevant part is called *external knowledge*. The relevant part is called *contextual knowledge*. At a given step, a part of the *contextual knowledge* is proceduralized. The *proceduralized context* is the part of *contextual knowledge* that is invoked, structured and situated according to a given focus (Figure 1). Proceduralization means that people use contextual knowledge in functional knowledge or causal and consequential reasoning. This proceduralization fulfills the need of having a consistent explicative framework to anticipate the results of a decision or action. This consistency is obtained by reasoning about causes and consequences in a given situation [14].

There are several views of context: context as conceptual drift (a context engine); context as a medium for the representation of knowledge and reasoning; context as what surrounds the focus of attention, etc. All these context concepts have been formalized and used in knowledge-based applications. However, these views are rather isolated. An analysis of shared context and its use in group work is also necessary. In the following section we present a framework that can be considered as a first step towards this goal.

### 3 Understanding Context in Group Work

A context may be seen as a dynamic construction with five dimensions: (1) time, (2) usage episodes, (3) social interactions, (4) internal goals, and (5) local influences [10]. Although the contextual elements in some situations are stable, understandable, and predictable, there are some situations where this does not occur. Cases having apparently the same context can be different. In order to reduce this, we use a conceptual framework whose objective is to identify and classify the most common contextual elements in groupware tools [18]. The goal of this framework is to provide guidelines for research and development in groupware and context.

According to McCarthy [11], the size of the contextual dimension is infinite. Thus, the framework considers only those contextual elements that are most relevant to task-oriented groups, i.e., contextual knowledge and proceduralized context [4]. The contextual information is clustered into five main categories: (1) people and groups, (2) scheduled tasks, (3) the relationship between people and tasks, (4) the environment where the interactions take place and (5) the tasks and activities that have already been completed. These clusters were borrowed from the Denver Model [19]. In synchronous environments, group members need to work at the same time; however, in asynchronous environments there might be a time lag between interactions. The needs of each type of environment are different, especially those that are related to contextual information [15].

The framework is a generic classification of contextual elements. It does not cover the peculiarities of a certain domain nor does it apply to a specific type of groupware. This generic framework is a starting point for a classification of contextual elements in specific domains, where new contextual elements may be considered relevant.

The first category provides information about the group members; it contains information about the individuals and the groups they belong to. The knowledge about the group's composition and its characteristics is important to be able to understand the potential ways in which the project or the task will be developed. This knowledge encourages interaction and cooperation [15]. This category is sub-divided into two types of context. The individual context carries information about each of the individuals who are members of a group. The group context data is similar to the aforementioned, but relates to the group as a whole. It includes the composition of the team, its abilities and previous experience as a group, and the organizational structure.

The second category provides information about scheduled tasks. Independently of how the interaction occurs, the group members need to be acquainted with the task characteristics. Task context is the name given to this context. Its goal is to identify tasks through their relevant characteristics: the task name, its description and goals, the deadline, the predicted effort, the technology, and other requirements.

The third category provides information about the relationship between the group members and the tasks. The goal of this category is to relate the action of each group member and the interaction s/he is involved in. This interaction begins with an execution plan, goes through a sequence of actions required to carry out the plan, and terminates when the task has been completed. If the interaction is interrupted before the task is completed, the reasons for the premature termination also form part of the context and are relevant to understanding the reason for the interruption.

This category is sub-divided into two types of contexts: the interaction context and the planning context. The interaction context consists of information that represents the actions that took place during task completion. When the interaction is synchronous, the details of the activity must be known at the time that it occurs. When the interaction is asynchronous, the overview of activities is what is most relevant.

The planning context consists of information about the project execution plan. This information can be generated at two different points. For ad-hoc tasks, the information appears as a result of the interaction. For scheduled tasks, it is generated at the time of the plan, i.e.; when the tasks are defined and the roles are associated to them. The planning context can include rules, goals, deadline strategies, and coordination activities.

The fourth category provides information on the environment. It represents the aspects of the environment where the interaction takes place. It covers both organizational issues and the technological environment; i.e., all the information outside the project (but within the organization) that can affect the way the tasks are performed. The environment gives further indications to group members about how the interaction will occur; for instance, quality control patterns are part of this context. This context also includes strategy rules, policies, financial restrictions and institutional deadlines.

The last category provides all the information about tasks that have already been completed. The goal of this category is to provide background information about the experiences learned either from the same group or similar tasks performed by other groups. It should include all contextual information about previous projects. The framework refers to this set of information as "historical context". This information is important for understanding errors and to be able to apply successful approaches from previous projects to current tasks. It can also be used out of the context of a project to

provide insight into working practices and team cooperation. A summary of the framework is shown in Table 1.

**Table 1.** Conceptual framework for the analysis of context in groupware [18]

Information type	Associated Contexts	Goals	Examples of contextual elements
Group Members	Individual (Synchronous & Asynchronous)	To identify the participants through the representation of their profiles.	<ul style="list-style-type: none"> <li>• Name</li> <li>• Previous experience</li> <li>• Working hours</li> </ul>
	Group (Synchronous & Asynchronous)	To identify the group through the representation of its characteristics	<ul style="list-style-type: none"> <li>• Members</li> <li>• Roles</li> <li>• Organizational Structure</li> </ul>
Scheduled Tasks	Task (Synchronous & Asynchronous)	To identify the tasks through the representation of their characteristics.	<ul style="list-style-type: none"> <li>• Goals, deadlines</li> <li>• Estimated effort</li> <li>• Activities</li> </ul>
Relationship between people and tasks	Interaction (Synchronous)	To represent in detail the activities performed during the task completion.	<ul style="list-style-type: none"> <li>• Exchanged messages</li> <li>• Presence Awareness</li> <li>• Gesture awareness</li> </ul>
	Interaction (Asynchronous)	To represent an overview of the activities performed during the task completion.	<ul style="list-style-type: none"> <li>• Artifacts generated</li> <li>• Activities completed <ul style="list-style-type: none"> <li>• Author</li> <li>• Results</li> </ul> </li> </ul>
	Planning (Synchronous & Asynchronous)	To represent the execution plan of the task to be performed.	<ul style="list-style-type: none"> <li>• Interaction roles</li> <li>• Rules</li> <li>• Strategies</li> <li>• Procedures</li> </ul>
Setting	Environment (Synchronous & Asynchronous)	To represent the environment where the interaction occurs; i.e., characteristics that influence the task execution.	<ul style="list-style-type: none"> <li>• Quality patterns</li> <li>• Policies</li> <li>• Financial constraints</li> <li>• Standard procedures</li> </ul>
Completed Tasks	Historical (Synchronous & Asynchronous)	To provide understanding about tasks completed in the past and their associated contexts.	<ul style="list-style-type: none"> <li>• Task Name</li> <li>• Versions of the artifacts</li> <li>• Contextual elements</li> <li>• Working Plan</li> </ul>

## 4 Context and Awareness in Groupware

Proceduralization of context involves the transformation of contextual knowledge into some functional knowledge or causal and consequential reasoning in order to anticipate the result of actions [16]. When people work as a group, context becomes especially relevant. Not only do individual contexts need to be proceduralized, but so does the group context. As described in the framework, group context is not simply the union or intersection of individual contexts. For instance, a specific person may work differently with a certain group of colleagues than with another one.

How is context processed when doing group work? Fig. 2 shows our proposed model. It is basically a *knowledge processing* procedure. People create knowledge individually. It is then communicated to the rest of the group as well as being presented in a User Interface (UI) and eventually stored. The *generation* step consists of a person contributing information to the group. This information could be contents

for the group’s output or it could be related information, such as questions, suggestions, or proposals. Part of this knowledge is stored to satisfy the condition “all contents must be saved”.

The *capture* step consists of procedures to gather some physical data from the generation step. For instance, in the case of joint text editing, the movement of the user’s mouse may serve as an indication of which part of the document the user is working on. In another example, a camera can capture the physical movements of a person; these movements might be important for another user, who may be wondering why the first person is not answering his/her questions.

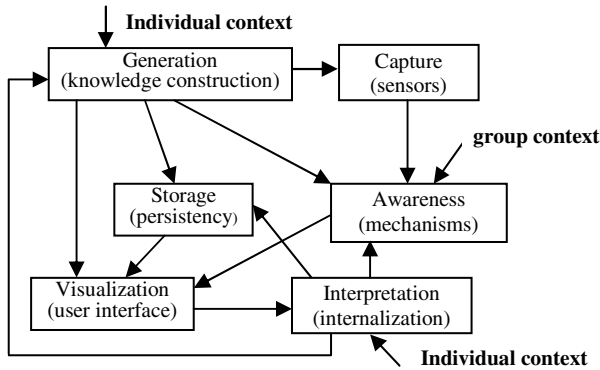


Fig 2. Context knowledge processing in group work [3]

The *awareness* step consists of the processing of information to be communicated to the other participants [9]. Note that it has several inputs. The first input is information from the generation step. An example would be a contribution that has just been written by a group member. This information needs to be transformed in some way, perhaps summarized or filtered to make it available to other people. In fact, this step takes into account the processing specifications given by individual users. Another type of input is from the capture step. Again, this information will probably be processed to avoid information overload. The awareness step also receives information from the storage step. This occurs, for example, when an agent decides to distribute a summary report on recent work in asynchronous systems. Finally, it should also be noted that there is a group context that is received as input. This represents important information that is needed to process the rest of the inputs.

The *visualization* step generates the user interface. It provides users with a physical representation of knowledge: icons, text, figures, etc. Input to this step can come from the generation procedure: the physical feedback a user receives when s/he contributes to the group.

Capture, storage, awareness and visualization are all processing steps that are performed by the system on the basis of users specifications and pre-established rules. Besides generation, there is another human processing step: the interpretation process. The person performs this step by visualizing the information and combining it with his/her individual context to transform it into knowledge. This is needed by the person to generate new contributions to the group and close the cycle of processing context. A person might need some information from storage and can request it; this petition

might be as simple as a mouse click on a button on the UI or it might be a complex query specification.

## 5 Contexts and Awareness in Practice

We use two groupware systems to illustrate the use of the framework and the contextual knowledge model: SISCO [2], a meeting preparation asynchronous system that is intended to support the group discussion occurring before an actual meeting; and CO2DE [12], a cooperative editor that handles multiple versions as a way to deal with conflicting views. Both systems support groups working with a common task. SISCO provides the organization of opinions about agenda items, and CO2DE provides one or more versions of a collaboration diagram in a software engineering project. Neither of the systems explicitly supports context, but they both use several context elements to support group work.

Notice that making context explicit is a way to remember, not only the way in which a solution was developed, but also the alternatives at the time of solution building, existing constraints, etc. Thus, awareness is achieved by comparing the context used at that time with the current context.

If the goal is to find a solution, it is also important to account for individual contexts. A specialist might propose a solution from his/her field of domain. Yet, another specialist may give constraints. In such a case, the first specialist will modify his/her context from the pair (problem, solution) to the triple (problem, context, solution). By working together, each person will be able to share more knowledge with the other members. Thus, their individual contexts will have a non-empty intersection, making their interaction short and efficient.

In SISCO, since the goal is to have a broad discussion, the selection is based on the contextual knowledge that each participant has about the meeting agenda items, as well as the diversity of individual contexts. The contributions are shared among group members to reduce repetitions and also to increase the quality of the contributions by making other participants' ideas explicit. This sharing promotes the internalization and idea generation processes. Since a repetition occurs when a person is working individually, the awareness step is dropped. The capture may still be needed, but it becomes trivial, and will probably just be presented on the UI.

SISCO must provide persistency of contributions to the discussion as well as awareness of the discussion contents. Whenever a member logs in, the system generates a schematic view of the discussion contents, indicating what is new to him/her. This keeps the contextual knowledge uniform among group members even when they have not connected to the system for long periods. Perhaps no one has complete knowledge of the contributions. Thus, the system must make contributions persistent and provide awareness mechanisms to allow users to update their individual contexts with the group context that are represented by the set of contributions.

The task context covers as much of the wide range of options and arguments related to the agenda items as possible. During the discussion, which is supported by SISCO using an IBIS-like argumentation model, most contributions are based on participants' individual context. Thus the authorship provides some hints about the associated context. SISCO also encourages participants to express not just their own

views, but to express those that are logically consistent with the task context. In this way, the system attempts to disassociate opinions from individual contexts and move them towards the task context. One way of achieving this is by removing authorship from the contributions.

Another way of supporting task context is through the definition of roles. When playing a role in SISCO, an individual is given a narrower context with specific awareness mechanisms. For instance, the coordinator role is provided with a *participameter*, a widget that informs about the level of participation in the discussion [1]. The *participameter* is considered a kind of group or task context and provides the coordinator with elements to decide on what to do. For example, when the participation level in a certain item is low the possible actions to be taken are: remind people, promote discussion, or even drop the item.

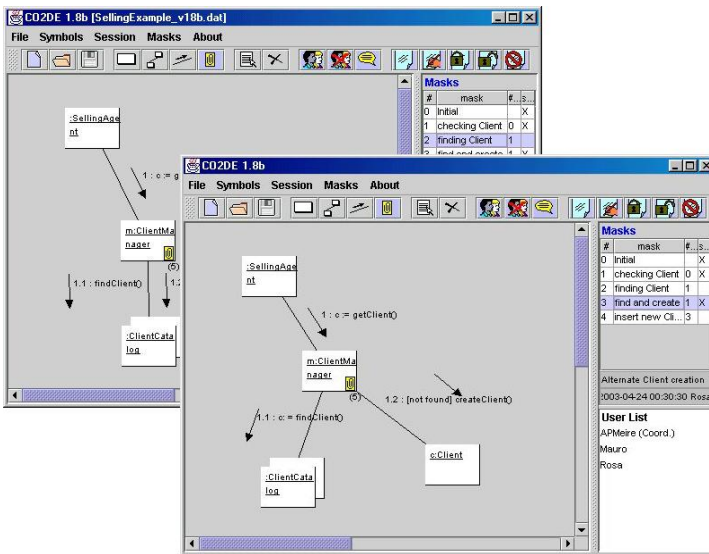


Fig 3. CO2DE user interface [12]

The CO2DE editor allows for individual contexts to be joined into a single diagram by providing a synchronous cooperative edition feature and a WYSIWIS interface (Fig. 3). Although this also allows asynchronous interaction, it does not focus on it. The diagram functions as the memory of the latest group context, which is the union of individual contexts. However, the context notion is not explicitly treated by CO2DE.

When conflicting views arise in a diagram, most cooperative editors encourages users to reach a consensus by means of a communication mechanism, e.g., a chat. CO2DE deals with conflicts in a different way. It allows several versions of the diagram to co-exist. It organizes the versions into a tree to associate each version to its origin, its alternative versions resulting from the conflict, and its further decomposition originated from another conflict. In none of these cases, however, does

the system represent contextual information; e.g., the conflict and the assumptions for a version. This information is kept within each individual context and is not stored.

During the elaboration of the diagram, several versions may co-exist. It is left to participants to solve the conflicts and express the resulting consensus in a single version. The CO2DE approach has the advantage of allowing users to represent their views in a more comprehensive format, since a single conflict usually involves several elements of the diagram. It is like discussing two or more options using the complete picture, instead of discussing each element one at a time. Another advantage is the representation of the work evolution by means of a set of step-refined versions. The approach also supports a mental comparison of two alternatives. With a simple click of the mouse the user can rapidly perceive the differences between diagrams.

The framework presented in this paper indicates a potential for improvement to CO2DE. When many versions of a diagram are present, it is desirable to have the rationale for each version stored with it, since even its creator may forget what it was. This context is not awareness information. The system should be extended to handle these explanations and allow the user to retrieve them by clicking on a specific button in the version representation. This is equivalent to the “requesting additional information” arrow from “Interpretation” to “Storage” in Figure 2.

## 6 Conclusions

The study of context and CSCW has largely been done independently. Perhaps this has not been beneficial for groupware designers, who might profit from research in contexts. This framework may be a first step in narrowing this gap by relating the concepts of context and groupware. The model representing how the awareness mechanism can carry contextual information illustrates how the notion of context is related to other widely used terms in CSCW, such as user interfaces, automatic capture, knowledge construction and storage.

The context process model presents group work as a knowledge-processing task that has some activities that can be performed by a machine as support to the human tasks. This dataflow-type modeling is novel. The presentation of context as knowledge flowing among different processing activities is also new.

The framework and the model can be applied together to obtain some insight into certain groupware designs. By considering context as knowledge that can be applied during group work, there can be a wider perspective than just focusing on the information provided to users by awareness mechanisms. Other groupware designs would probably be suitable for analysis and improvement from this viewpoint.

## Acknowledgements

This work was supported by grants from: CNPq (Brazil) PROSUL AC-62, Fondecyt (Chile) No.1040952. Marcos R. S. Borges was partially sponsored by a grant from the Secretaria de Estado de Educación y Universidades of the Spanish Government.



## References

1. Borges, M.R.S., Pino, J.A.: Awareness Mechanisms for Coordination in Asynchronous CSCW. Proceedings of the Workshop on Information Technology and Systems. Charlotte, NC, USA (1999) 69-74
2. Borges, M.R.S., Pino, J.A., Fuller, D., Salgado, A.: Key issues in the design of an asynchronous system to support meeting preparation. *Decision Support Systems*, Vol. 27 No. 3. Elsevier (1999) 271-289
3. Borges, M.R.S., Brézillon, P., Pino, J.A., Pomerol, J-Ch.: Bringing Context to CSCW. Proceedings of the 8th International Conference on Computer Supported Cooperative Work in Design Vol II. Xiamen, China, IEEE Press (2004) 161-166
4. Brézillon, P.: Individual and team contexts in a design process. Proceedings of the Hawaii International Conference on System Sciences (HICSS-36). Hawaii, USA, IEEE Computer Society Press (2003) CD-R
5. Brézillon, P.: Using context for Supporting Users Efficiently. Proceedings of the Hawaii International Conference on System Sciences (HICSS-36), Hawaii, USA, IEEE Computer Society Press (2003) CD-R
6. Brézillon P.: Contextualizations in a Social Network. *Context in Social networks and virtual communities*. *Revue d'Intelligence Artificielle*, Vol. 19 No. 3. Hermès Science Publications, (2005) 575-594
7. Context 2005: <http://www.context-05.org/> Accessed on 28 January 2005
8. Dourish, P.: Seeking a Foundation for Context-Aware Computing. *Human-Computer Interaction*, Vol. 16 No. 2-4. Lawrence Erlbaum (2001) 87-96
9. Dourish, P., Bellotti, V.: Awareness and Coordination in Shared Workspaces. Proceedings of the Computer-Supported Cooperative Work Conference. ACM Press (1992) 107-114
10. Greenberg, S.: Context as a Dynamic Construct. *Human-Computer Interaction*, Vol. 16 No. 2-4. Lawrence Erlbaum (2001) 257-268
11. McCarthy, J., Notes on formalizing context, Proceedings of 13<sup>th</sup> International Joint Conference on Artificial Intelligence. Chambéry, France, Morgan Kaufman (1993) 555-560
12. Meire, A., Borges, M.R.S., Araujo, R.: Supporting Collaborative Drawing with the Mask Versioning Mechanism. Proceedings of the 9<sup>th</sup> International Workshop on Groupware. *Lecture Notes in Computer Science* Vol. 2806. Springer-Verlag, Berlin Heidelberg New York (2003) 208-223
13. Moran, T.P. and Dourish, P.: Context-Aware Computing. *Human-Computer Interaction*, Vol. 16 No. 2-4. Lawrence Erlbaum (2001) 87-94
14. Naveiro, R., Brézillon, P., Souza, F. : Contextual knowledge in design: the SisPro project. *Espaces Numériques d'Information et de Coopération*, Vol. 5 No. 3-4. C. Simone, N. Matta & B. Eynard (Eds.). Hermès Science Publications (2002) 115-134
15. Pinheiro, M.K., Lima, J.V., Borges, M.R.S.: A framework for awareness support in groupware systems. *Computers in Industry* Vol. 52 No. 1. Elsevier (2003) 47-57
16. Pomerol J-Ch., Brézillon P.: Dynamics between contextual knowledge and proceduralized context. *Lecture Notes in Artificial Intelligence* Vol. 1688. Springer-Verlag, Berlin Heidelberg New York (1999) 284-295
17. Rittenbruch, M. ATMOSPHERE: A Framework for Contextual Awareness, *Int. Journal of Human-Computer Interaction* Vol. 14 No. 2. Lawrence Erlbaum (2002) 159-180
18. Rosa, M.G.P., Borges, M.R.S., Santoro, F.M.: A Conceptual Framework for Analyzing the Use of Context in Groupware. Proceedings of the 9<sup>th</sup> International Workshop on Groupware. *Lecture Notes in Computer Science* Vol. 2806. Springer-Verlag, Berlin Heidelberg New York (2003) 300-313
19. Salvador, T., Scholtz, J., Larson, J.: The Denver Model for Groupware Design, *SIGCHI Bulletin* Vol. 28, No. 1. ACM Press (1996) 52-58

# Grid Authorization Management Oriented to Large-Scale Collaborative Computing

Changqin Huang<sup>1,3</sup>, Zhiting Zhu<sup>1</sup>, Xianqing Wang<sup>2,3</sup>,  
and Deren Chen<sup>3</sup>

<sup>1</sup> ECNU-TCL Joint Workstation for Postdoctoral Research on Educational Technology,  
East China Normal University, Shanghai, 200062, P.R. China

<sup>2</sup> Guangdong Institute of Technological Personnel, Zhuhai, 519090, P.R. China

<sup>3</sup> College of Computer Science, Zhejiang University, Hangzhou, 310027, P.R. China  
cqhuang@zju.edu.cn, ztzhu@dec.ecnu.edu.cn, xqwang\_kgy@126.com

**Abstract.** In this paper, we propose Subtask-based Authorization Service (SAS) architecture to fully secure a type of application oriented to engineering and scientific computing. We minimize privileges for task by decomposing the parallel task and re-allotting the privileges required for each subtask. Community authorization module describes and applies community policies of resource permission and privilege for resource usage or task management. It separates proxy credentials from identity credentials. We adopt a relevant policy and task management delegation to describe rules for task management. The ultimate privileges are formed by the combination of relevant proxy credential, subtask-level privilege certificate and community policy for this user, as well as they conform to resource policy. To enforce the architecture, we extend the RSL specification and the proxy certificate, modify Globus's gatekeeper, jobmanager and the GASS library to allow authorization callouts, and evaluate the user's job management requests and job's resource request in the context of policies.

## 1 Introduction

Grid computing [1] has been widely accepted as a promising paradigm for large-scale wide-area distributed computing in recent years. One goal of Grids is to provide easy and secure access to the Grid's diverse resources. Grid infrastructure software such as Legion [2] and Globus [3] enables a user to identify and use the best available resource(s) irrespective of resource location and ownership. However, realizing such a pervasive Grid infrastructure presents many challenges due to its inherent heterogeneity, multi-domain characteristic, and highly dynamic nature. One critical challenge is providing authentication, authorization and access control guarantees. As Grids move from an experimental phase to production facilities, the security issue of a Grid application becomes more imperative.

Engineering and scientific computing is a typical problem suited for being solved in grid environments. This type of task is commonly either computation-intensive or data-intensive, the problem granularity is widely large and computational tasks are often long-lived. It need be divided into many subtasks, and then be distributed to

many relevant nodes and run in parallel way. The issue needs not only fine-grained authorization for resource usage and management but also fine-grain authorization for task management: This requires giving one or a group of its members in relevant VO [Virtual Organization] the apt privileges to manage any jobs using VO resources by policies. Privileges that are not properly allocated/delegated to a unit of subtask may interfere with the normal progress of the task or increase security threats by the more disclosure of the privileges.

The Globus toolkit is the most popular grid environment and de facto grid standard, however its current security services are yet poor. Globus has adopted the Grid Security Infrastructure (GSI) [4] as the primary authentication mechanism. According to GSI, the simple authentication is performed and the resource allows the task to use all privileges of the user, similarly, to subtasks run in parallelism, Globus deals with all privileges of subtask irrespective of the privilege difference among its subtasks. In Engineering and Scientific Computation Oriented grid environment, it violates commonly the least privilege principle [5]. This coarse-grain authorization for task management is not suitable for the requirement of actual authorization for task management in long-lived grid application.

In this paper, we focus on the security requirements posed by engineering and scientific computation applications in grid. This paper is organized as follows: Section 2 reviews related work in the arena of grid security. In section 3, the proposed authorization architecture and overall policy are described. Section 4 describes the current implementation of the architecture within Globus. Related considerations are presented in Section 5. Finally, conclusions and future work to be addressed in Section 6.

## 2 Related Work

In recent years, the related researches are making great progress. I. Foster et al. [4] provide the basis of current grid security: “grid-map” mechanism, very limited support for fine-grain authorization decisions on grid requests. Mapping grid entities to local user accounts at the grid resources is a common approach to authorization. A grid request is allowed if such a mapping exists and the request will be served with all the privileges configured for the local user account. However, this authorization and access control mechanisms are not suitable for flexible authorization decision.

L. Pearlman et al. [6] propose the Community Authorization Service (CAS) architecture. The CAS server is designed to maintain authorization information for all entities in the community and grants restricted GSI proxy certificates (PCs) to community members. Access control system enables multiple owners and administrators to define fine-grained usage policies in a widely distributed system. It reduces overhead by separating the administration of resource specific issues from those that are community specific. Drawbacks of this approach include that the approach of limiting the group’s privileges violates the least-privilege principle and that it does not consider authorization of task management.

W. Johnston et al. [7] provide grid resources and resource administrators with distributed mechanisms to define resource usage policy by multiple stakeholders and make dynamic authorization decisions based on supplied credentials and applicable usage policy statements. In Akenti system, proposed by W. Johnston et al. the appli-

cable access policy for a specific request results from the intersection of use-policies defined by the resource owners and the group and role assignments made in attribute certificates by user managers. It presents fine-grained and flexible usage policies in a widely distributed system defined in the Akenti policy language. Fine-grained access decisions based on such policies and user attributes are enforced by the application code, and it does not consider authorization issue of task management.

R. Alfieri et al. [8] present a system conceptually similar to CAS: the Virtual Organization Membership Service (VOMS), which also has a community centric attribute server that issues authorization attributes to members of the community. In VOMS, however, the subjects authenticate with their own credentials and the attributes allow for the use of community privileges. M. Lorch et al. [9] give the same architecture, called PRIMA. Except that in PRIMA the attributes are not issued by a community server but rather come directly from the individual attribute authorities, PRIMA and VOMS have similar security mechanisms. And they do not consider authorization issues of task management, and they only support the creation of small, transient, ad-hoc communities without imposing requirements to deploy group infrastructure components like community servers.

Besides the typical paradigms mentioned above, G. Zhang et al. [10] present the SESAME dynamic context-aware access control mechanism by extending the classic role based access control (RBAC) [11]. SESAME complements current authorization mechanisms to dynamically grant and adapt permissions to users based on their current context. But, monitoring grid context in time is high-cost. S. Kim et al. [12] give a WAS architecture to support a restricted proxy credential and rights management, which uses workflow to describe the sequence of rights required for normal execution of a task in order to minimize rights exposure. It does not consider the actual conditions of large-scale task running at many nodes in a parallel way. K. Keahey et al. [16] describe the design and implementation of an authorization system allowing for enforcement of fine-grained policies and VO-wide management of remote jobs. However, it does not specify and enforce community policies for resource usage and management currently. M. Lorch et al. [17] enable the high-level management of such fine grained privileges based on PKIX attribute certificates and enforce resulting access policies through readily available POSIX operating system extensions. Although this mechanism enables partly the secure execution of legacy applications, it is mainly oriented to collaboration computing scenarios for small, ad hoc groups.

### **3 Subtask-Based Authorization Service (SAS) Architecture**

#### **3.1 SAS Architecture**

SAS architecture is concerned with the different privilege requirements of subtasks, user privilege for the resource and resource policy in virtual community, and task management community policy and authorization delegation. So SAS architecture includes three functional modules as shown in Fig 1.

To minimize privileges, the parallelizable task is decomposed and the least privileges required for each subtask is re-allotted. This contribution is described in the part of subtask-level authorization module in Subsection 3.2. Community policy based authorization mechanism is addresses in the next. To apply a flexible task

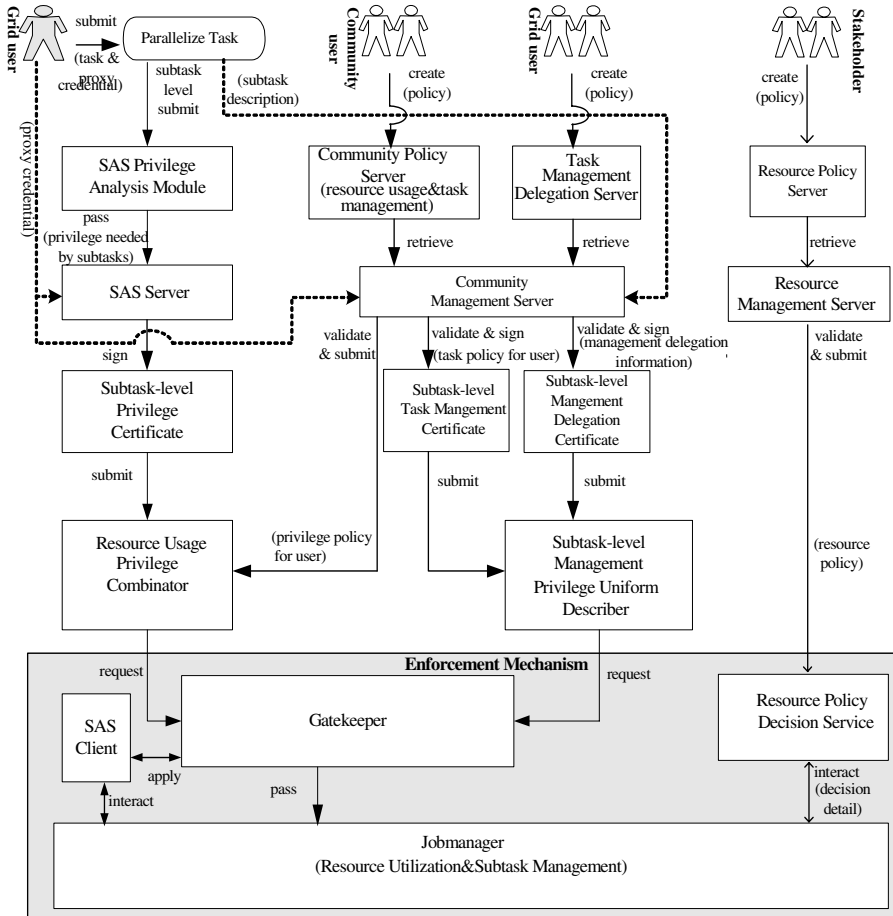


Fig. 1. SAS architecture overview

management, a delegation mechanism collaboratively performs the authorization delegation for task management together with a relevant management policy. Its details exist in the part of task management authorization module in the subsection followed.

### 3.2 Privilege Management and Overall Authorization Policy

**Subtask-level authorization module** concentrates on minimizing privileges of tasks by decomposing the parallel task and analyzing the access requirement. To further conform to the least-privilege principle, a few traditional methods restrict the privileges via the delegation of users themselves rather than the architecture, moreover, they only restrict privileges of a whole specific task, not for its constituents (such as subtask). In an engineering and scientific computation application a task is usually large-scale and long-lived. It needs to be divided into many subtasks, and then be distributed to many relevant nodes and run in parallel. Whilst, even for one single task, privileges required by its distinct subtasks may differ according to operations of

these subtasks. By the parallelization and analysis of the task, SAS can obtain the task's subtasks and relevant required privileges: subtask-privilege pair. The privileges indicate the way how associated subtask access to resources at certain nodes. Each task has a subtask-level privilege certificate for recording subtask-privilege pair. An example of this certificate is shown in Fig. 2. To prevent malicious third party from tampering with a subtask-level privilege certificate, a trusted third party (a SAS server) signs the certificate. To control a subtask's process and verify the subtask-level privilege certificate, the SAS client will run with GRAM during resource utilization.

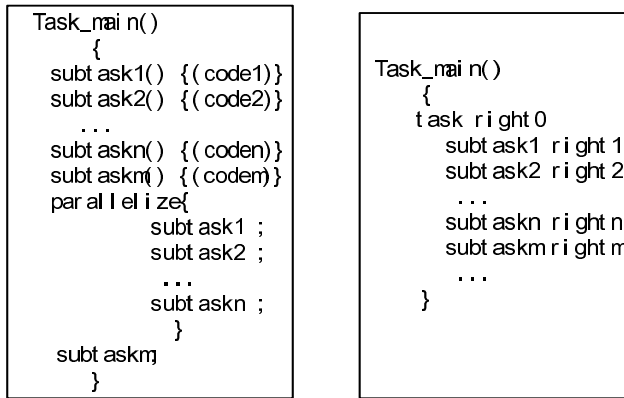


Fig. 2. An example of task and subtask-level privilege certificate

**Community authorization module** addresses community authorization mechanism for community member. In Globus, “grid-map” mechanism is conducted, but it is neglected that grid collaboration brings out common rules about privilege for resource usage, resource permission, and so forth. The SAS architecture imposes mechanisms similar to the CAS with a combination of traditional grid user proxy credential and CAS, and both resource usage policies and task management policies are added into the community policy server. By a further extension, the stakeholders of resource can create themselves resource permission policies into a resource policy server. Two trusted third parties (a community management server and a resource management server) and two policy servers (a community policy server and a resource policy server) are exercised. The first two servers have capabilities to validate the conditions and sign associated policies, as well as manage the respective policies. A community management server is responsible for managing/signing policies that govern access to a community's resources (Actually, task management policies in the next module are also dealt with by it), and a resource management server is responsible for managing/signing the policies that govern resource permission to grid users. Two policy servers store the community policies and resource policies, respectively. The ultimate privileges of resource usage are formed by the subtask-level privilege certificate and the policy for this user, and the actual rights need to accord with resource policy by resource policy decision service module during policy enforcement. The policy

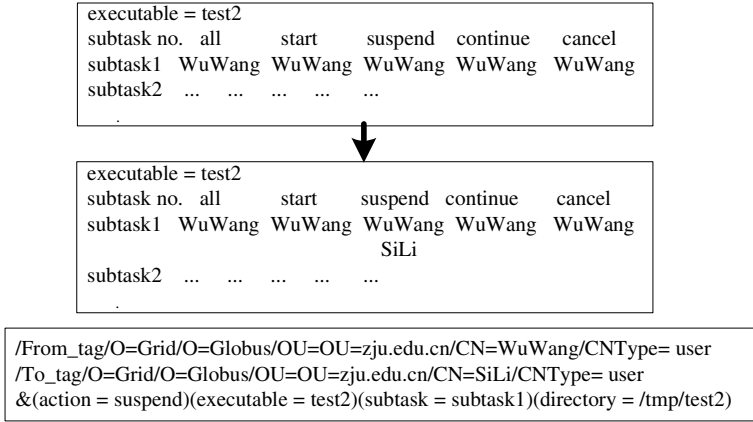
servers are built or modified by community administrators or certain specific resource administrators/stakeholder.

In community policy based authorization, if a grid user submits one parallelized task, an associated community management server will come into operation; heretofore, the community policy must have been created by a certain community administrator or other specific community users. The community policy server stores all community policies. At runtime, the community management server retrieves all policies associated with this grid user, after relevant analysis, this server presents two types of policy, i.e. resource usage policies and task management policies. Then, it validates the proxy credential and signs usage privilege policies for this user. During it, only task management policies is formed a specific certificate (subtask-level task management certificate), which will be sent to the subtask-level management privilege uniform describer in order to form the actual privileges for task management; the other policies are submitted to the resource usage privilege combinator, in which they are composed into a subtask-level privilege certificate matched with original grid user's task, and conduct enforcement via a request for gatekeeper.

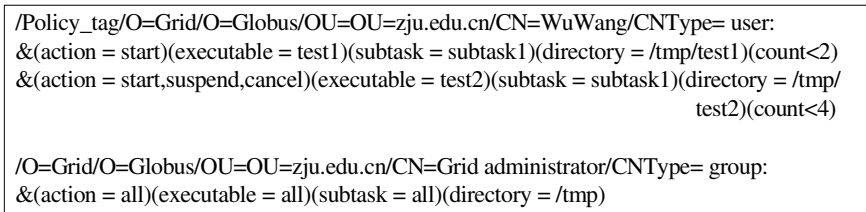
In the part of resource (permission) policy based access control, stakeholders of resources create the relevant policies; during enforcement mechanisms, according to the requirement of grid task being running, the relevant policies are retrieved, validated and signed by the resource management server. The resource policy decision service accepts resource policies and implements their enforcement.

**Task management authorization module** is responsible for managing privileges of task management, authorization to task management and related works. In dynamic grid environments, there are many long-lived tasks. Users may also start jobs that shouldn't be under the domain of the VO. Since the user who has submitted the original job may not always be an optional manager, the VO wants to give a group of its members the ability to manage any task using VO resources. This module imposes a community task management policy and task management delegation to describe rules of task management, and both of the two mechanisms are beneficial to flexible task management in an expressive way. Community task management policies denote the rules of task management in the whole community, and they are stored at the community policy server, where resource usage policies are put. Task management delegation server records a variety of management delegation relations among community users. Once the delegation relation is formed, this task will be able to be managed by the delegated user at runtime. In addition to these functionalities described in the previous subsection, a community management server is responsible for authenticating task management delegation. To keep compatibilities, the subject list only consists of the owner of the task by default. Fig. 3 shows an example for a task management delegation and the change of the subject list for task management privilege to this delegation. Subtask-level management privilege uniform describer produces the task management description expressed by extending the RSL set of attributes. An example of task management description is shown in Fig. 4.

Except that task management policies are implemented as described in the previous sub-section, task management delegation is completed as follows. Firstly, some grid users create their delegation assertions, and sent to task management delegation server, which record all delegation relations for grid users in the VO. When any



**Fig. 3.** A task management delegation and the relevant change of the subject list



**Fig. 4.** An example of task management description

grid user submits a request for task management via his proxy credential, the community management server retrieves all delegation assertions associated with the current management request. After relevant validation with the proxy credential, it finds out the assertions suitable for subtasks belonging to the current requested task, and then signs another specific certificate (subtask-level management delegation certificate), which will be sent to the subtask-level management privilege uniform describer in order to combine this certificate and the subtask-level task management certificate into the actual privileges for task management. Finally, all task management requests are sent to gatekeeper to implement these authorizations according to this certificate.

## 4 Enforcement Mechanisms

Enforcement of fine-grained access rights is defined as the limitation of operations performed on resources or tasks/subtasks by a user to those permitted by an authoritative entity. Based on the Globus Toolkit 2.2, SAS architecture implements the subtask-level authorization, flexible task management, and context-aware authorization.



## 4.1 Specification Extensions

**Proxy credential extension:** In the SAS architecture, the rich information of authorization must be efficiently carried to relevant points, and the carrying facilities should be suitable for information parsing and semantic combining (such as privileges combining). It is our selected approach that all authorization information is uniformly carried by the proxy credential. To realize grid resource utilization or grid task management in the Globus, a grid user's request usually carries both the contents of his request and corresponding proxy credential granted by this grid user, and the proxy credential is the basis of GSI delegation mechanisms for grid authorization, so extending the proxy credential is a straightforward method in implementation mechanisms for the SAS. The proxy certificate, which is a passport of proxy credential, is one kind of X.509 certificates[13], and there exist a fine-defined extension field in X.509 version 3. By defining extensions to X.509 certificates to carry community policies and management delegation assertions, subtask-level task management certificate and subtask-level management delegation certificate can be produced; similarly, subtask-privilege pairs are recorded into the extension field like a type of attribute, so subtask-level privilege certificate is formed. Because the privilege policy for resource usage is similar with the privilege of subtask-privilege pairs, we have not made the former have an independent certificate, but archive it into subtask-level privilege certificate at the point of the privilege combinator. Subtask-level management delegation may happen independently, for example, a common grid user, who is neither an original grid user nor a community member, requests for task management. For this scenario, a subtask-level task management certificate can be formed. Similarly, there exist cases of no subtask-level management delegation certificate. So subtask-level task management certificate and subtask-level management delegation certificate must coexist in our implement. In proxy credential, GSI also treats the policy and assertion as opaque, meaning that the creator of a policy and resources enforcing it need to understand it.

**RSL extension:** All types of certificate formats are designed for legibility to enforcement mechanisms. Therefore the actual policy language or other description language should be the generalized specification system, whilst it meets certain scalability to support arbitrary policy languages (such as ASL [14]) in the future. In our specification we do not state a specific language to be used, but instead extend Globus's Resource Specification Language (RSL). In virtue of the flexible authorization certificate, this allows us to evolve our policy language over time as new requirements are understood, and as policy languages themselves evolve.

In the SAS architecture, by introducing new tags and declarations for RSL, the system implements these descriptions of subtask-privilege pair, community resource policy, community management policy, management delegation and resource usage policy, and also conducts a multi-value policy. As shown in Fig. 2, we refer to the parallelized task description, such as the MPI program (MPICH-G2) is applied in SAS), and use "Task\_main()" to indicate the begin of subtask-privilege pair; We let "Policy\_tag" followed by "User-Resource-Privilege" pair denote resource usage privilege based on community policy, and let "Policy\_tag" followed by "User-Task-

Privilege” pair denote task management privilege based on community policy as described in Fig. 4. As shown in Fig. 3, we introduce “From\_tag” and “To\_tag” to denote the delegation source and the delegation target, respectively. Finally, the resulting task delegation presents the similar form of task management privilege based on community policy. In the language of resource policy for access control, we let “Policy\_tag” followed by “Resource-Permission” pair denote the beginning of these policy declarations.

In addition to these tags mentioned above, we introduce many expressive declarations to describe the actual contents, which are usually put after these tags in relevant certificates or the others.

## 4.2 Enforcement Mechanisms in Underlying Systems

Subtask-privilege pairs, community policies and delegation assertions are carried in extended proxy credentials. They are successfully sent to the final part of authorization enforcement, where restricted resources are utilized under restricted privileges or tasks are managed. On one hand, we modify Globus’s gatekeeper and jobmanager to collaborate with our other enforcement facilities, on the other hand, we build two independent modules, that is, the SAS Client and the Resource Policy Decision Service. Finally, Globus’s GASS library is extended in order to communicate with certain additional modules.

As a lot of information mentioned above (such as subtask-level privilege certificate) needs to be validated for authentication and integrity, we develop a callout API for validation at gatekeeper, which integrates with the OpenSSL toolkit, so we apply the OpenSSL toolkit’s verifying module when the SAS Client requests for authentication. Afterwards, the system must evaluate and conduct the policies and assertions. We have developed a callout API and library that parsing certificate and controlling resource usage and task management, and internally our implementation uses the Generic Authorization and Access control (GAA) API [15], which make us have better choices in the future because the GAA supports new policy languages. Here are its other operations: After the authentication is passed, the SAS client interacts with jobmanager via the callout API. The callout passes to the SAS client all the information relevant to access control, such as the credential of the user requesting a remote job, the action to be performed (such as starting or canceling a job), a unique job identifier, and the job description expressed in RSL. The SAS client performs its functional operations and responds through the callout API with either success or an appropriate authorization error. This call is made whenever an action needs to be authorized. Whilst, we modify gatekeeper to allow one user to signal a jobmanager instance owned by another user, and extend the GRAM protocol to let gatekeeper or jobmanager return authorization errors with reason indications. For much communication happens between the grid remote client and grid server, so Globus’s GASS library is extended to provide corresponding supports.

However, there exist many differences in the part of resource policy based access control, i.e. resource permission policy. The resource policy decision service module itself applies the OpenSSL toolkit to authenticate the stakeholder signature, and then interact with jobmanager via a callout API and relevant library developed by us.

However, we only implement its main functions depending on the access control of operating systems at time being.

## 5 SAS Related Considerations

In the SAS architecture, the exposure of the user privileges and the security holes of the entire community can be reduced because the grid user must use the subtask-level privilege certificate corresponding to the parallelized task. Certainly, implementing the useful functionalities brings forth a few of issues to be considered.

1. For the proposed fine-grained authorization, some overhead will occur. In the SAS architecture, the SAS privilege analysis module creates a file of subtask-privilege pair from a grid user program, and the SAS server module signs its subtask-level privilege certificate. The community management server retrieves relevant information and signs it. During associated final enforcement, the gatekeeper and the SAS client must collaborate authenticate certificates, and the SAS client must load the certificates and conduct the corresponding limitations by the additional toolkit and jobmanager. As well, these other modules, such as resource policy, are also complex. We have found that they produce much overhead not only in time but also in resources.

On one hand, the SAS architecture is oriented to engineering and scientific computation, in which tasks are usually large-scaled and long-lived, so the above cost of time is less in contrast with the running time of these tasks. Whilst, the absolute cost has been well reduced: A large part of the submitted programs can be parallelized and present themselves as MPI codes, therefore the main module, SAS privilege analysis module, will easily exercise at the little cost of time; In the SAS architecture, the signed certificates are made by setting the necessary information automatically, and all policies and delegation assertions are submitted by friendly graphical interfaces. Therefore, the overhead from grid users can be obviously reduced. All authentication and enforcement of authorization are completed with a few classical toolkits or techniques (such as OpenSSL, and GAA package), which are well compliant with our system, so some expenses are also lessened.

On the other hand, the expenses of resource are well controlled. The above modules really occupy some grid resources, such as memories and CPU cycles. However, this overhead is not a fatal factor in a large-scaled grid enabled application because Grid is the proposed environment to support the task that requires computing resource at the level of the supercomputer or great amounts of other computers, so the additional expenses from our modules are acceptable. Most tasks have simple repeatable operations and many subtasks are identical. According to these characteristics, some occupied resources can be shared.

2. The security issues of proposed modules have been considered. An attacker could perform his task by making the subtask sequence of the malicious executable similar to the original sequence. In the SAS, every third authentication server will sign the relevant certificates or files after recording the task's unique identifier, original user and current user. Finally, all these information will be validated at runtime. Thereby, a malicious executable cannot be executed for no passing the validations.

## 6 Conclusions and Future Work

To decrease security threats and improve security mechanism flexibility, Grid authorization architecture should fully have “separation of concerns” and “least privilege access” mechanism. In this paper, we propose Subtask-based Authorization Service (SAS) architecture to fully secure a type of application oriented to engineering and scientific computing. This type of task is widely large-scale and long-lived. It need be divided into many subtasks run in parallel way. These subtasks may require different privileges. In SAS architecture, subtask-level authorization module minimizes privilege for task by decomposing the parallel task and analyzing the privilege requirement for each subtask, All/a part of entire privileges for task execution are allotted to its subtasks in the form of a subtask-level privilege certificate. Community authorization module considers the necessity of grid community rules about resource permission and privilege for resource usage and task management. As a result, it enforces three types of policies. It separates proxy credentials from identity credentials. Task management authorization module adopts a task management policy and task management delegation to describe rules for task management. The ultimate privileges are formed by the combination of relevant proxy credential, subtask-level privilege certificate and policy for this user, as well as they conform to resource permission policy. To enforce the architecture, we extend the RSL specification and the proxy certificate with an extension field, modify Globus’s gatekeeper, jobmanager module and the GASS library to allow authorization callouts, and evaluate the user’s job management requests and job’s resource request in the context of policies defined by the resource owner and VO.

Currently, SAS architecture is prototype, and only simple task is tested. So SAS architecture should be improved in practice, a complex task is needed to check it. At the same time, accounting for resource usage and subtask secure migrating will be studied in the near future.

## Acknowledgement

The work presented in this paper is supported by Scientific Research Fund of Hunan Provincial Education Department, China (Grant No. 04A037), and the National High-Tech. R&D Program for CIMS, China (Grant No. 2002AA414070).

## References

1. Foster, I., Kesselman, C. and Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations, *International Journal of Supercomputer Applications*, 15(3) (2001) 200-222
2. Grimshaw, A., Wulf, W.A., et al.: The Legion Vision of a Worldwide Virtual Machine, *Communications of the ACM*, 40(1) (1997) 39-45
3. Foster, I. and Kesselman, C.: Globus: a metacomputing infrastructure toolkit, *International Journal of Supercomputer Applications*, 11(2) (1997) 115-128

4. Foster, I., Kesselman, C., Tsudik, G., Tuecke, S.: A Security Architecture for Computational Grids, Proc. of 5th ACM Conference on Computer and Communications Security Conference, (1998)
5. Salzer, J.R. and Schroeder, M.D.: The Protection of Information in Computer Systems, Proc. of the IEEE, (1975)
6. Pearlman, L., Welch, V., et al.: A Community Authorization Service for Group Collaboration, Proc. of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks, (2002)
7. Johnston, W., Mudumbai, S., et al.: Authorization and Attribute Certificates for Widely Distributed Access Control, Proc. of IEEE 7th International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, (1998)
8. Alfieri, R., et al.: VOMS: an Authorization System for Virtual Organizations, Proc. of the 1st European Across Grids Conference, (2003)
9. Lorch, M. Adams, D.B., et al.: The PRIMA System for Privilege Management, Authorization and Enforcement in Grid Environments, Proc. of the 4th International Workshop on Grid Computing, (2003)
10. Zhang, G. and Parashar, M.: Dynamic Context-aware Access Control for Grid Applications, Proc. of the 4th International Workshop on Grid Computing, (2003)
11. Sandhu, R., Coyne, E. et al.: Role-based Access Control Models, Proc. of the 5th ACM Workshop on Role-Based Access Control, (2000)
12. Kim, S., Kim, J., Hong, S. et al.: Workflow-based Authorization Service in Grid, Proc. of the 4th International Workshop on Grid Computing, 2003.
13. Tuecke, S. et al.: Internet X.509 Public Key Infrastructure Proxy Certificate Profile. 2002.
14. Jajodia, S., Samarati, P., Subrahmanian, V.S.: A Logical Language for Expressing Authorizations. Proc. of IEEE Symposium on Security and Privacy. (1997)
15. Ryutov, T. and Neuman, C.: Access Control Framework for Distributed Applications, IETF Internet-draft draft-ietfcat-acc-cntrl-frmw-05.txt, (2000)
16. Keahey, K., Welch, V. et al.: Fine-Grain Authorization Policies in the Grid: Design and Implementation, Proc. of 1st Intl Workshop on Middleware for Grid Computing, (2003)
17. Lorch, M. and Kafura, D.: Supporting Secure Ad-hoc User Collaboration in Grid Environments, Proc. of the 3rd IEEE/ACM International Workshop on Grid Computing, (2002)

# Research on Network Performance Measurement Based on SNMP

Shufen Liu, Xinjia Zhang, and Zhilin Yao

College of Computer Science and Technology, Jilin University,  
Changchun, Jilin 130012, P.R. China  
jlluren@163.com, {liusf, yaozl}@jlu.edu.cn

**Abstract.** This paper presents our recent research on network performance measurement. From various perspectives, our work focuses on how to obtain the measuring data using the Simple Network Management Protocol (SNMP). Several models are proposed to measure the network performance effectively. These models are then evaluated and validated through implementations and analysis. The results show advantages of the proposed approach for network performance measurement based on SNMP and potential applications in telecommunication domain.

## 1 Introduction

Performance measurement is the network management foundation, through which we can acquire all kinds of data that related to the network performance index. By analyzing these data and studying the current network status, we can master network behavior rules, find out any problems and solve them on time to ensure an acceptable network performance.

Performance measurement is a very complex task. Modern computer network is a huge, integrated architecture, which contains countless heterogeneous hardware and software elements, including devices, links and application services that associate with each other.

Devices are the most basic network components. They can be divided into two types: network devices and host devices. The index of device performance measurement mainly includes usability, CPU utilization rate, memory utilization rate and file system space utilization. A network device needs to be measured of its interface parameters also, such as usability and bandwidth.

Link is the connecting entity between devices in the computer network. Its performance measurement index mostly includes throughput, usability rate and utilization rate. We also need to measure delay, jitter and packet loss rate of the end to end connection.

Application service is the operation runs in the computer network. Application service measurement mainly includes response time, usability and throughput. Since there are various kinds of application services with complex indexes and methods, we only discuss the key application problems of performance measurement. In particular we will discuss how to obtain measurement data by using SNMP protocol<sup>[1][2][3]</sup>.

## 2 Usability

Usability is one of the most important indexes of network performance. It represents the usable time of computer network system (including devices and links.) and application services. It can be expressed by the rate of every year, every month, every week, every day or every hour network runtime to corresponding time quantum.

For example, a network can provide service 24 hours a day, 7 days a week. So if it can run 165 hours in 168 hours of a week, the usability is 98.21%.

In general usability is related to redundancy, which is a solution of usability goal but not the goal of network construction. Redundancy is to add multiple links or standby devices to avoid network service interruptions and to reduce network loads.

Usability is associated with reliability, but is more explicit. Reliability is a diverse question including precision, error rate, stability, and time between failures.

Usability is also associated with resiliency. Resiliency means how much pressure can network endure and how fast can network resume from errors.

### 2.1 Theoretical Model

Usability is based on the reliability of a single system in the network. Reliability, which can be popularly expressed by MTBF (Mean Time Between Failures), is the probability of a system executing its special function under the special condition at special time. System usability can be expressed as:  $A = \text{MTBF}/(\text{MTBF} + \text{MTTR})$ . MTTR is mean time to repair after failures appeared.

If we consider the expected system load, the calculation of usability would become quite complex. For example, suppose there are  $n$  multiplexing devices in a system, each link can handle the peak load by ratio  $q$ . When the ratio of service requests does not reach peak is  $p$ , the system can use  $k(k < n)$  devices to handle all the stream loads. However, when service requests reach the peak, all the links have to be used to handle the stream loads, but can only handle part of them, the ratio is  $r$ . The system usability can be expressed as:

$$Af = \sum_{i=0}^n (\text{performance of use } i \text{ links}) \bullet P[\text{use } i \text{ links}]. \quad (1)$$

$P[\ ]$  means “the probability of  $[\ ]$ ”.

Probability of using  $k$  links is:

$$P_{\text{coms}} = A^k \bullet (1-A)^{n-k}. \quad (2)$$

“ $A$ ” means the probability of using any link.

When the system can handle all the stream loads by using  $k$  links, each link handles stream loads of ratio  $q_a$ , then:

$$Af = \sum_{i=0}^k \min(1, q_a \bullet i) \bullet P[i \text{ links}]. \quad (3)$$

When the system has  $n$  links to work, but can only handle a part of the stream loads, the ratio is  $r$ , and each link can handle the stream loads of  $q_b$ , then:

$$Af = \sum_{i=0}^n \min(r, q_b \cdot i) \cdot P[i \text{ links}]. \quad (4)$$

Thus, the entire usability is:

$$Af = P \cdot \sum_{i=0}^k \min(1, q_a \cdot i) \cdot P[i \text{ links}] + (1 - P) \cdot \sum_{i=0}^k \min(r, q_b \cdot i) \cdot P[i \text{ links}]. \quad (5)$$

As a matter of fact, network is very complex. It is hard to evaluate the total system usability in measurement due to the complex topology structure, difference between devices and continuous and dynamical changes of routing. The usability measurement can be simplified to measure each physics link.

## 2.2 Implementation Method

There are two modes to measure network devices and links. They are aggressive detecting mode and passive accepting message mode. Aggressive detecting mode is used commonly. It includes ICMP ping and SNMP poll.

By using ICMP ping, the management station sends an ICMP message to an agent. If the agent is unusable, the management station would return an overtime message. In real time measurement, we send ten 100 bytes ICMP messages simultaneously every 5 minutes to the target device. In this way, we ensure the accuracy of measurement without occupying too much network bandwidth. We can use following formula to calculate:

$$Usability = \frac{\text{Total measure times} - \text{invalid times}}{\text{Total measure times}}. \quad (6)$$

In order to enhance the efficiency and accuracy of measurement, and to avoid destination unreachable circumstances, it is necessary to use distributed architecture. The distributed architecture is to adapt closeness principle to set a management station for each network segment. Each management station is in charge of network devices performance monitoring of its local area. The network segments are divided according to location of the huge WAN. Time and data of management stations should be synchronized, i.e., using a consistent clock. Measuring results are stored in the central database. Since the management station is close to the measured devices, the results of measurement will be more accurate. When using SNMP poll, we can get the interface status precisely. But we need to use vast samplings to measure its usability.

Aggressive detecting mode is suitable to measure whether a device can be used or not at a certain time. But as to usability statistics, which uses sampling method, it is hard to avoid missing to detect some exceptions. For example, a device is useable in sampling point time "t", and is also usable at time "(t+5)". The measuring program would consider this device to be usable from "t" to "(t+5)". However, when the device is restarted between "t" to "t+5", we cannot find out this situation by using the aggressive detecting mode. For improvement we can shorten the sampling interval, but it cannot solve the problem completely. Relatively, the passive accepting message mode is absolutely predominant in this aspect.



Passive accepting message mode mainly adopts SNMP trap mode. We runs a demon process (called management process) on the management station, and let the process listen at some UDP port (62 as a default port). Then we make correlative configuration at the managed device. When the device is shutdown, or is restarted, or its port is down, the device will send a trap to the management station. When the management station receives the trap, it will standardize its format, and then store the trap into the database. We can use the following formula to calculate interface usability:

$$Usability = \frac{Total\ time - Sum\ of\ all\ DOWN\ state\ time}{Total\ time}. \quad (7)$$

In this way, we can get all the information about the device and whether its ports can be used or not. With the combination of the distributed acquisition structure, we can reduce the disintegrated data for losing UDP packages. Relatively, it is an effective usability measurement solution.

By comparing the realization of above usability measuring methods, we can use SNMP trap as the basic measurement method, and use SNMP poll and ICMP ping as complementary methods, to measure the usability of devices and ensure data integrity, consistency and reliability.

### 3 Response Time

There are two types of response time: network response time and application response time. Network response time is a network layer concept, which represents the data flow transmission time between two nodes. When response time is abnormal, namely it exceeds a certain threshold, it might reflect that the network is either congested or has other problems.

In general, we should emulate end users' behaviors as much as possible when we measure the response time. For example, when a user opens a browser, inputs a Web address, presses the enter key, from this point on, till the page shows up in the browser, the whole process is the response time of the Web application for this user at this specific time. It includes the response time from the user's computer to Web server and the handling time of Web server to the page request. To add them together, we get the application response time.

But, it might be extremely difficult for us to measure the application response time, since we do not have the appropriate tools, and the users are isolated and uncountable. In addition, it is unnecessary to measure application response time to solve the network performance problem or to expand capacity in the future, even though from the application performance measurement point of view, it is necessary to measure it.

#### 3.1 Theoretical Model

We can use a dedicated device, as well as ICMP (such as ping, traceroute) to measure response time. Even if we cannot get application level response time, we can understand that how the network hops and how fast it transmits IP packets<sup>[4]</sup>. For example, we can use ping command to get time delay from the management station to

a key node in the network, such as an interface of a core router, or an access device of SP (service provider), or an important user station, and so on. But this method cannot reflect response time from a user device to its destination device. It can only collect and report response time from the management station to a user device.

Another method other than centralized detection is distributed detection, which means taking the time from a user access point to its destination device as the response time. If the device that the user accessed is a Cisco device implementing Service Assurance Agent (SAA)<sup>[5]</sup>, by which we can measure the time from the router to any IP device, you can not only simulate ICMP protocol packet to detect response time of the appointed IP device, but also can run simulations by sending packets of UDP, TCP, DNS, DHCP, SMTP, FTP, HTTP, VOICE, etc, to emulate users' behaviors and measure related response time. The data obtained by this method are only approximate, and do not accurately reflect the response time. However, the method is effective.

In the network with QoS queue rules, we can use Cisco's Service Assurance Agent to emulate data flow and measure the time period from one end to another in the network effectively.

## 4 Precision

The concept of precision is whether or not the network device has transmitted messages without faults. It can be expressed by the percentage of packets transmission with no fault to total packets transmission in a period of time. For example, if an interface has 2 fault packets while sending 100 packets on average, the error rate is 2%, while the precision is 98%.

As for the earlier network, especially the wide area network, the error rate in a certain level is acceptable. However, with the development of high speed network and more demands on critical network services, the network transmission must be steady and reliable. Many technical documents indicate the following reference values of error rate: The typical threshold of analog link is  $10^{-5}$ ; The error rate of data optical link is about  $10^{-11}$ ; copper line link error rate is about  $10^{-6}$ . In shared Ethernet, errors are usually evocated by collision, and the frame influenced by legal collision should not exceed 0.1%.

Any error beyond the error rate may evoke lots of problems, such as network performance reduction, network service interruption, and users' complains.

Some common interface errors include: nonstandard cable system; electronic disturbance; software or hardware defects.

So it is necessary to monitor the error rate that must be corrected in measurement. It can not only point out discontinuous malfunction link that must be corrected, but also can point out the appearance evidence of noise or disturbing source. As a result, we can prevent any network interruption as early as possible.

### 4.1 Theoretical Model

Precision is based on network performance endurance. When error rate of a network device interface rises to exceed a certain original threshold value, it triggers an event

to inform the management station, which should ensure whether the interface has problem or not, and whether it is acceptable or not. If it can be accepted, we should lower the threshold, repeat the procedure until unacceptable error rate appears. The threshold at that time is the benchmark data of interface precision measurement.

In this paper we suppose that the threshold is predefined, and it does not need measurement adjustment.

Any measurement data exceed threshold can be treated as evidence of performance alert analyzing.

The error rate formula in percentage format is:

$$ErrorRate = \frac{\Delta ifInErrors \bullet 100}{\Delta ifInUcastPkts + \Delta ifInNUcastPkts}. \quad (8)$$

The accuracy formula in percentage format is:

$$Accuracy = 100 - \frac{\Delta ifInErrors \bullet 100}{\Delta ifInUcastPkts + \Delta ifInNUcastPkts}. \quad (9)$$

$\Delta x$  means the difference between the results of twice checking the variant “x”, as follows. IfInErrors means the count of error packets of input interface data; IfInUcastPkts means count of the unicasting packets of input interface data; ifInNUcastPkts means the count of multicasting packets of input interface data.

## 5 Utilization Rate

Utilization rate refers to the resource usage situation in a given period of time, usually expressed by the percentage of usable resource to the full capacity.

The most common usage of utilization ratio is to find out potential bottleneck or blocking area. This is very important, because the response time increases by the power exponent with resource utilization ratio. If we cannot find and deal with the barrage in time, we may not be able to control it. Thus, it may bring down the network performance. In addition, measurement of utilization rate can find resources that have low utilization rate or cannot be fully used.

### 5.1 Interface Utilization Rate

When a barrage appears in an interface, the messages would be arranged into the interface queue. If the queue is full, the messages would be discarded. For example, when we transfer data from a fast interface to a slow one, package loss may occur. When a message is discarded, upper layer protocols would probably require sending messages repeatedly. If we lost many messages, there will be lots of re-sending message flows in the network. In this way, network link would be blocked. Through utilization ratio measurement, we can find out problems in time, then carry out necessary load balancing or route adjustment to avoid network paralysis<sup>[6]</sup>.

So, the measurement of network interface utilization rate is very important. There are two calculating formulas as follows. We should choose one of them to calculate

the interface utilization rate according to whether the tested connection is semi-duplex or full duplex.

Semi-duplex connection interface utilization rate calculating formula is as follows:

$$Utilization = \frac{(\Delta ifInOctets + \Delta ifOutOctets) \bullet 8 \bullet 100}{Interval\ seconds \bullet ifSpeed} \tag{10}$$

ifInOctets means the number of inflow data bytes;  $\Delta ifInOctets$  means the difference between two successive checkings of a set interval, i.e., the inflow data in the time interval;  $\Delta ifOutOctets$  means the difference between two successive checking of a set interval, i.e., the outflow data in the time interval; ifOutOctets means the number of outflow data bytes; ifSpeed means the speed that the interface handles and delivers the packets, as follows.

As for the full duplex, utilization rate measurement is different. For example, a T-1 serial sequence connection line speed is 1.544Mbps, which means it can send and receive data at the same time at the speed of 1.544Mbps, and the whole bandwidth is 3.088Mbps. So we can use the larger one of inflow and outflow bytes to calculate utilization rate, as follows:

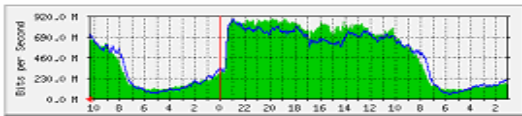
$$Utilization = \frac{Max(\Delta ifInOctets, \Delta ifOutOctets) \bullet 8 \bullet 100}{Interval\ Seconds \bullet ifSpeed} \tag{11}$$

We ignore the flow direction factor and cannot obtain accurate results by using this method. We can use a more accurate method to calculate by detaching inflow and outflow, i.e.:

$$Utilization_{in} = \frac{\Delta ifInOctets \bullet 8 \bullet 100}{Interval\ Seconds \bullet ifSpeed} \tag{12}$$

$$Utilization_{out} = \frac{\Delta ifOutOctets \bullet 8 \bullet 100}{Interval\ Seconds \bullet ifSpeed} \tag{13}$$

Daily Chart (5 Min Average)



Max In : 899.8 Mb/s (90.0%) AVG In : 481.7 Mb/s (48.2%) Now In : 661.4 Mb/s (66.1%)  
 Max Out : 891.0 Mb/s (89.1%) AVG Out : 449.2 Mb/s (44.9%) Now Out : 651.9 Mb/s (65.2%)

Fig. 1. Inflow and outflow of the interface

The above formulas seem to be simple. We do not consider the specific QoS protocol. But the real time measurement calculation showed that they are accurate and reliable not only to LAN but also to WAN interfaces.

To the most of physical links, interface utilization rate is the link utilization rate. So it is also an effective link utilization measurement method to use SNMP poll to

obtain interface using situations. Figure 1 is the real time measured inflow and outflow.

### 5.2 CPU Utilization Rate

Some key routing functions, such as protocol analysis, packet exchange transaction are carried out in memory by sharing CPU. If CPU utilization rate is too high, the routing table might not be able to be updated, packets might be lost and the network performance might be seriously affected.

We can use Cisco device as an example to explain how to obtain CPU utilization rate by using SNMP.

First, we should understand the corresponding management variables of CPU utilization rate. In Cisco management information base, there are two tables including this information. One table is OLD-CISCO-CPU MIB (Or OLD-CISCO-SYS MIB); the other is CISCO-PROCESS MIB<sup>[8]</sup>.

As to a single CPU system, we can obtain CPU utilization rate from response variables in the above two tables; but for multiple CPU system we can only obtain information from the second table. Figure 2 is the real time measured CPU utilization rate.

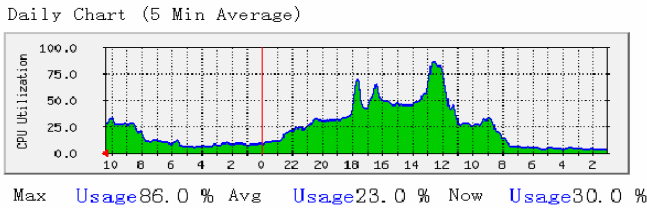


Fig. 2. CPU utilization rate

### 5.3 Memory Utilization Rate

Monitoring memory usage can help us to find out memory leakage and abnormal network events. If a process applies for memory block, but does not release it after using it, we call it “memory leakage”. If this kind of situation happens all the time, it would exhaust all memory and make the system collapse. If there is no enough memory, we would be unable to allocate buffer for other processes, like the routing table memory requests. Eventually the system performance would be affected.

We can use SNMP poll to get the value of the management variable, and then use the following formula to calculate the CPU utilization rate:

$$Utilization = \frac{ciscoMemoryPoolUsed}{ciscoMemoryPoolUsed + ciscoMemoryPoolFree} \tag{14}$$

ciscoMemoryPoolUsed means the bytes of memory that are used in the memory pool; ciscoMemoryPoolFree means the bytes of memory that are unused in the memory pool. Figure 3 is the real time measured memory utilization rate.

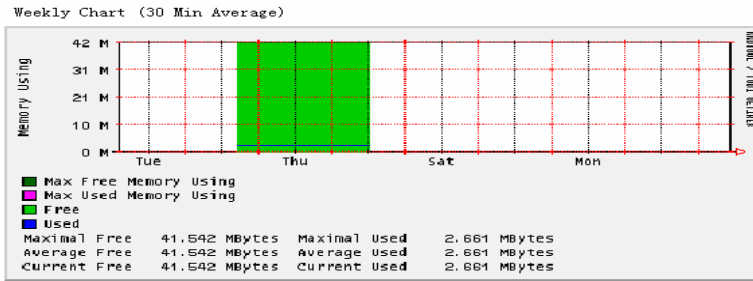


Fig. 3. Memory utilization rate

## 6 Conclusion

Network performance measurement is an important aspect of network management. In this article, we compare and analyze different measurement methods to different indexes of network performance. Our work focuses on how to obtain the measuring data using the Simple Network Management Protocol (SNMP). These models are then evaluated and validated through implementations and analysis. The results show advantages of the proposed approach for network performance measurement based on SNMP and potential applications in telecommunication domain.

## References

- Harrington, D., Presuhn, R.: An Architecture for Describing SNMP Management Frameworks, RFC 3411, <http://www.ietf.org/rfc> (2002) 5-6
- Case, J., Harrington, D.: Message Processing and Dispatching for the Simple Network Management Protocol (SNMP), RFC 3412, <http://www.ietf.org/rfc> (2002) 19-32
- Presuhn, R., Case, J.: Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP), RFC 3416, <http://www.ietf.org/rfc> (2002) 4-23
- White, K.: Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations, RFC 2925, <http://www.ietf.org/rfc> (2000) 5-10
- Network Monitoring Using Cisco Service Assurance Agent, [http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgr/ffun\\_c/fcfcprt3/fcf017.htm](http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgr/ffun_c/fcfcprt3/fcf017.htm).
- Gac, R.J. Jr, Harding, D.Jr, Weiser, J., Traffic and trend analysis of local and wide area networks for a distributed PACS implementation, Proceedings of SPIE Volume 3980, (2000) 447-457
- Presuhn, R., Case, J.: Management Information Base (MIB) for the Simple Network Management Protocol (SNMP), RFC 3418, <http://www.ietf.org/rfc> (2002) 2-20.
- Song, S., Huang, J.: Internet router outage measurement: An embedded approach, 2004 IEEE/IFIP Network Operations and Management Symposium: Managing Next Generation Convergence Networks and Services, NOMS, 1 (2004) 161-174

# Concepts, Model and Framework of Cooperative Software Engineering

Yong Tang, Yan Pan, Lu Liang, Hui Ma, and Na Tang

Dept. of Computer Science, Sun Yat-sen University,  
Guangzhou 510275, P.R. China  
issty@zsu.edu.cn, pianoll@tom.com

**Abstract.** Recently, the cooperation aspect of distributed teamwork in software engineering has become a hot research topic. This paper first reviews the concepts of cooperative software engineering. Then, a process model for cooperative software engineering is investigated, which forms the starting point for the analysis, structuring, management and synchronization of cooperative software development tasks. Next, universal design principles for an environment to support cooperative software development projects are obtained. The identified design principles serve as the basis for the development of the model. Finally, a typical framework for cooperative software engineering environment is proposed and its key components are described.

## 1 Introduction

Nowadays, it is a trend that more and more software development projects involve a large number of developers. In conventional software engineering, the efficiency of teamwork is generally improved by the coordination among team members, which can be achieved by exchanging formal documents and providing support for keeping these documents consistent. However, as the number of developers in a project increases, the potential communication among team members may increase dramatically. Bandinelli et al. [1] pointed out that due to the cooperative nature of software development, success is dependent upon “the quality and effectiveness of the communication channels established within the development team”. In order to support teamwork more effectively, it is also important to move the focus beyond coordination towards cooperation. However, a survey of current research approaches and development environments reveals that group-supportive aspects often fail to meet the requirements for efficient support of communication and coordination [2]. Although some conventional tools, such as electronic mail, workflow management systems, synchronous debugging tools and group editors, do support group work in software projects, they are generally not integral components of a development environment or they only support limited activities. In order to achieve the success of software development based on distributed, cooperative and group work, several platforms or environments for supporting cooperative software engineering have been proposed [3–6]. However, there is a strong need to make a more precise description of cooperative software development process and its related model, framework and integrated environment as the precondition for conducting cooperative software engineering.

## 2 The Definition of Cooperative Software Engineering

### 2.1 Implication of “Cooperation”

In software development process, cooperation needs coordination processes to harmonize tasks among development team members. In turn, coordination needs corresponding communication mechanism to exchange information among the teammates. Therefore, cooperation should be achieved on the basis of coordination and communication.

**Cooperation** is the manner of coordination that is necessary for agreeing on common goals and for the coordinated achievement of common work results among the participants [7].

Cooperation usually implies shared goals among different actors [8]. Bischofberger et al. [6] identified two forms of cooperation: policy-driven cooperation and informal cooperation. Policy-driven cooperation is achieved by exchanging and handling well-structured documents effectively, and guaranteeing correct concurrency access to artifacts. This kind of cooperation usually takes place when team members encounter objective and technical troubles during the development process, and is often executed in predefined workflows and formalized processes. On the other hand, informal cooperation is characterized by the unrestricted exchange of structured or unstructured information, and the mutual influence among the members who carry out a task collaboratively. Such cooperation enables the co-workers to build mutual understanding of cooperative work, meaning that a member may be influenced by the others' subjective thought in the environment of freely exchanging ideas.

### 2.2 Two Dimensions of Software Engineering

Generally, Software engineering can be regarded as a sort of collection of product-centric activities and process-centric activities. Based on literature studies as well as our own experience and understanding from practical software developments, the two activities can be described as follows:

**Product-centric activities** mainly comprise two parts: requirement specification and system development procedure. The related artifacts include source codes, executable programs, illustrative documents, configuration documents and user manuals. The production processes are based on continuous development stages in which a number of predefined intermediate artifacts may be produced.

**Process-centric activities** usually aim at achieving the segmented software products. They mainly focus on the communication, interaction, coordination and cooperation among software developers in the whole team. They are based on the computer network and a set of assistant tools. More clearly, these activities consist of process management, coordination mechanism, product management, quality monitor, security control, version control and risk assessment. For the purpose of assuring intermediate results, reference lines are introduced into software engineering, defining and supervising project states that the developers and users have agreed upon for synchronization of their cooperative work processes [7,9].



### 2.3 The Definition of Cooperative Software Engineering

Based on the description of cooperation and two perspectives of software engineering in previous sections, cooperative software engineering can be defined as follows [3,6,9,10]:

**Cooperative Software Engineering** is the mutual operation, collaboration and cooperative work among developers on the basis of distributed computer network. It comprises all kinds of software engineering methods, norms and tools that support teamwork flexibly and effectively. It covers formal and informal communication and coordination requirements within a software development process that is necessary for the planning, execution and coordination of spatially and temporally distributed activities and tasks. Accordingly, cooperative software development also encompasses both product-centric and process-centric activities on the part of all participants whose common goal is the accomplishment of a software product.

## 3 Characteristics of Cooperative Software Engineering

**Task-Related Distribution:** Due to the large scale and increasing complexity of software engineering, it is often required to decompose a project into many subprojects, as well as to specialize within a development organization extending beyond project boundaries [9].

**Spatial and Temporal Distribution:** Spatially distributed teams are able to handle different subprojects sequentially or in parallel during the lifecycle of the project. Various project goals could be achieved at different times by the synchronization or coordination of work processes.

**Interaction:** In accordance with the two distributions described above, the construction of a suitable communication infrastructure is also needed to cover the communication, coordination, cooperation and other requirements. The interactions among team members could be either synchronous or asynchronous, depending on the specific requirement itself.

**Dynamic Development:** As various uncertainties always lead to frequent changes in the schedule of software development as well as the assignments among co-workers and the states of resources, developers are suggested to track all kinds of the dynamic information to guarantee the consistence of diverse work processes.

**Concurrent Development:** Activities are coordinated in parallel rather than sequentially. Co-work loops with constant communications are much shorter. The functional team and product realization processes are able to react to changes more quickly. The developers have to think about all the factors during the lifecycle, from the conception design to the project implementation and update. In addition, lower product cost can be achieved by executing effective decisions which are made early in the software development processes.

## 4 Research Fields Related to Cooperative Software Engineering

During the last decades, how to better assist the realization of cooperation in software development has attracted more and more attention. Significant contributions have been made to this research field.

**Process-Centered Software Engineering (PCSE):** Ever since the introduction of Process-Centered Software Engineering (PCSE), a series of process models have been applied in many software projects, because they are able to achieve cooperation in a way of describing a specific software process with all the activities and information flows it comprises. PCSE tries to establish a comprehensive theoretical basis for the purpose of understanding, describing and enhancing specific software processes [11]. The resulting process model consists of the description of objects (software artifacts and other process data) that are expressed as a set of rules defining the preconditions, operation processes and conclusions of various activities. However, PCSE ignores some other factors in software engineering, such as creativity, uncertainty, informal or interactive cooperation etc. as they are free from the standardized workflows and formalized processes [11].

**Workflow Management (WFM):** According to WfMC's definition [12], WFM systems are driven by the formalized workflow descriptions, and are obviously similar to cooperative software engineering environment in terms of supporting cooperative software development. WFM controls and manages a user's task sequence by a set of predefined operations or steps in business work, which may be sequential, parallel or interlaced. Once a workflow is precisely defined in a process-oriented WFM environment, it is usually impossible to escape from the modeled sequence of cooperative work steps at run time. In other words, WFM cannot perfectly meet the requirements of high dynamic or high flexible cooperation.

**Computer Supported Cooperative Work (CSCW):** The cooperative development of software project is a kind of cooperative work to some extent, causing that the core principle of Cooperative Software Engineering could also be regarded as an evolution or extension of CSCW [10].

The goal of CSCW is to assist real users to achieve the consistence of cooperative activities. A CSCW system consists of four elements: role, shared object, cooperative activity and cooperative event. Role describes the responsibility of a member in cooperative work. Shared object is the common object operated by co-workers during cooperative processes. Cooperative activity describes the specific and segmented cooperative process. Cooperative event, which is used to harmonize the actions of team members, is the indicator of how the cooperation goes on and how the state changes. CSCW strives to support both synchronous activities and asynchronous informal interactions. The environment for developing a CSCW system comprises a set of APIs (Application Programming Interface) supporting cooperative work, based on which developers can select suitable cooperation models and control mechanism to easily build a new software project.

**Concurrent Engineering (CE):** The concept of CE was brought forward almost at the same time (1986) as CSCW did. CE, a systematic approach regarding system

integration and parallel design, also concentrates on collaborative teamwork. Its technologies for supporting cooperation are highly relative to the research of CSCW.

**Computer-Aided Software Engineering (CASE):** CASE aims at the implementation of semiautomatic or automatic software development process. However, restricted by themselves, most CASE tools can only satisfy the requirements of collaboration and cooperation by using special design methods in special working environments. In other words, these tools lack of the all-purpose capability. In a study examining the support of collaborative facilities in CASE tools, Vessey and Sravanapudi [13] pointed out that available CASE tools provide only a few features that are commonly found in groupware systems.

### 5 A Processes Model for Cooperative Software Engineering

Based on the two-dimension characteristics of software engineering (described in Section 2.2), Altmann and Pomberger [9] proposed a model (shown in Figure 1) for cooperative software engineering to describe the development activities along with the associated project team members and the relationships among the workflows as well.

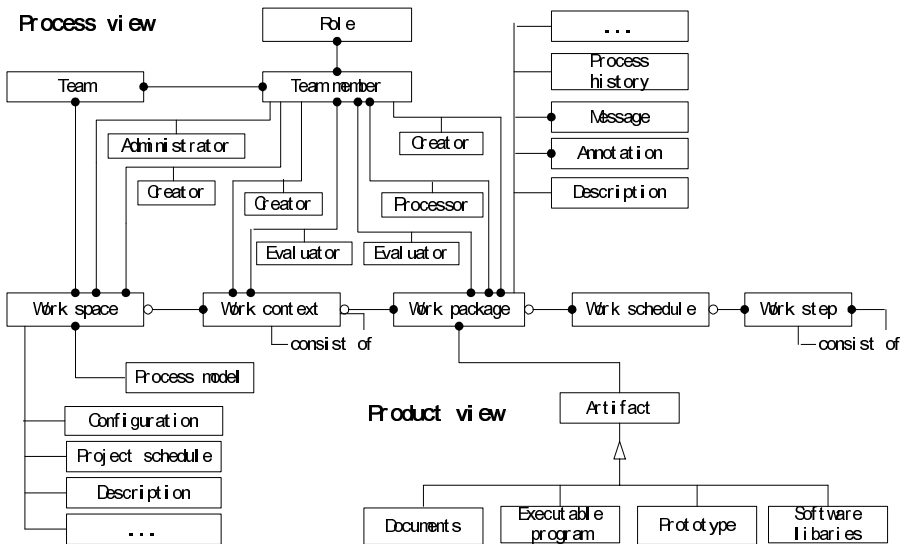


Fig. 1. Model of cooperative software development processes

The model comprises two logically connected areas:

- **Process view:** The kernel of this model is a set of components that describe the tasks of development activities and relationships among them. The process-related actions of a software development project occur when an overall task is decomposed into multiple smaller subprojects and subtasks. Each individual

subtask is assigned to a corresponding individual process participant or a small team.

- Product view: The product view includes the results of a software project. These development results consist of documents, executable programs, prototypes and software libraries.

## 6 Environment for Cooperative Software Engineering

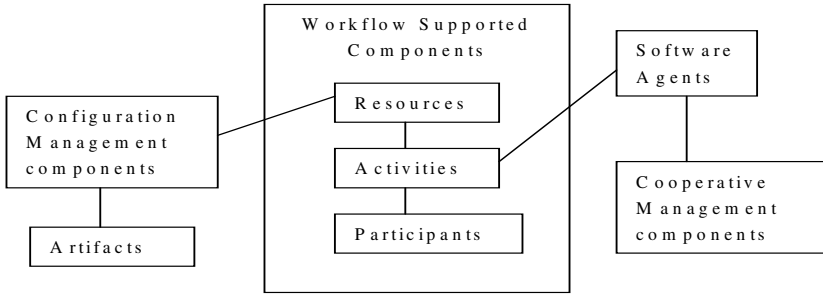
Cooperative Software Engineering should depend on applications of suitable methods, tools and environment to support cooperation. A good engineering environment can shorten the cycle of system development and reduce product cost, and let developers well apply the process models in cooperative work processes.

In order to identify the requirements for cooperative software engineering environment, Altmann and Weinreich [14] have made a deep research in the following aspects: the way software development teams work together, the kind of information exchanged, the underlying needs for coordinating the work of software development teams, and the kinds of existing project groups etc. As a result, they summarized a list of specific requirements:

- Individual- and group-specific views of current development process
- Group-, project- and organization- specific information
- Context-dependent document templates, check lists and other guidelines
- Predefined word procedures, each of which contains a list of tasks representing the standard or outline case of work performance
- Status and history information about an ongoing project and its software artifacts
- Browsing facilities to visualize and sort an activity sequence according special aspects
- Configuration of project-specific constraints and regulations
- Automatic notification of team members about changes and modifications
- Mechanisms for supporting awareness in cooperative development activities
- Communication independency of time and location
- Asynchronous informal communication (where dispersed team members can jointly comment and annotate documents)

## 7 A Typical Framework for Cooperative Software Engineering

So far, several platforms for supporting cooperative software engineering have been proposed [2~6,15]. These platforms have similar components to meet the common requirements of cooperative software engineering. We classify the user needs of cooperative software development into three main parts: software process control, artifacts management, communication and coordination. In accordance with these needs, a typical cooperative software engineering framework should generally have several key components as follows (Figure 2):



**Fig. 2.** A typical framework of cooperative software engineering

### 7.1 Workflow Supported Components

Workflow technologies can be used to model software development process. In workflow models, a software process is decomposed into many activities, resources and participants, for operating pre-planned activities always achieve great efficiency. In general, workflow supported components in cooperative software engineering environment mainly have the following three types:

**Process Modeling Components:** A process model provides the formalized description of software processes in computers. It defines all kinds of activities, participant roles, resources and run-time parameters in software processes, as well as the control flows and data flows in activities. There are many workflow modeling methods proposed to model software processes, such as Petri nets, object models, language arts theory, directed graphs and so on [16]. Good process modeling mechanisms should have good description ability, good flexibility and good support for further modification and dynamic evolution. Additionally, some extensions to workflow model, such as temporal extension [17,18], have been put forward to enhance the description ability of workflow models. However, special visualized modeling tools are required to support such workflow modeling.

**Process Analysis and Verification Components:** The purpose of modeling software development processes is to ensure their smooth execution and the expected business objectives. Thus, it is required for these models to be analyzed and verified by some process analysis and verification mechanisms before they are deployed in run-time environments. There are three types of process analysis [19]:

- Validation analysis is used to verify whether the execution of processes can achieve expected business objectives.
- Correctness analysis is used to verify the correctness of process models.
- Performance analysis is used to evaluate some performance parameters of process models, such as average execution duration of processes, average waiting time of activities, average using rate of resources and so on.

**Process Engine:** Providing well-defined practices for software specification, continuous software process improvement has become one of the key factors of software development. Software processes should encompass quality strategies, document

schemes and predefined work procedures which are looked as guidelines for planning and carrying out cooperative activities. Therefore, in cooperative software engineering environment, a Process Engine is needed to control the execution flows and trigger the process steps according to the pre-defined workflow model.

## 7.2 Configuration Management Components

As it can assist in organizing and controlling various artifacts produced during the lifecycle of software development, Configuration Management has been widely recognized as one of the key components in traditional software engineering. Tools supporting configuration management have evolved into sophisticated software environments, in order that they can enhance the ability of modeling and managing the states of all kinds of related items in software development processes. Therefore, Configuration Management is also very important in cooperative software engineering environments. It not only provides helps in synchronizing or merging parallel versions of software artifacts, but also supports efficient and effective coordination over long distances and low bandwidths.

## 7.3 Cooperation Management and Related Interaction Mechanism

Cooperation Management focuses on the two aspects of cooperation (as described in Section 2.1): formal cooperation which occurs in standardized workflows and development processes, and informal cooperation which occurs freely from the rigid spatial and temporal assignments in various workplaces and tasks. Because the latter is difficult to be modeled in workflow models, cooperation agents and related cooperative rules are introduced. Each activity needing cooperative actions is represented by a cooperation agent, and these agents interact with each other according to the predefined cooperative rules in Cooperation Manager. Cooperation management tools mainly support the team in planning collaborative work, provide to-do lists and informal communication, allow the delegation of work packages (clusters of small sub-activities that are assigned to team members) to other users or groups of people involved in task performance, notify changes, and provide up-to-date overviews of work progresses [14].

There is a need to establish some mechanisms in cooperation agents for communicating and exchanging messages with each other. Basically, synchronous mechanism and asynchronous mechanism have played an important role in supporting cooperation:

**Synchronous Mechanism:** A requirement for achieving cooperation among team members is that all the members should be provided a consistent working environment. The generation of cooperative actions must comply with a certain time-order, which is created and managed by the synchronization mechanism. In other words, synchronization mechanism manages the time-order of all kinds of cooperative events that occur in cooperative processes. The difficulty of well designing a synchronization mechanism is to describe synchronous relationships and provide real-time services.

**Asynchronous Mechanism:** The informal cooperation (described in Section 2.1) focuses on the shared small pieces of information that cannot be formalized (e.g.,

ideas and short term plans) and is not suitable to be put into activities with fixed structures. Therefore, when work teams are dispersed over long distances, a developer may be frequently interrupted by requests for some specific information that nobody else can provide. To support such cooperation and decrease the number of interruptions, an asynchronous mechanism should also be included into a cooperative software engineering environment.

## 8 Conclusions

The development of a large-scale and complicated software project is characterized by spatial and temporal distribution involving many developers. In order to improve the efficiency of development process and ameliorate the quality of software products, cooperative work mechanisms are necessary to be introduced into software engineering.

Cooperative software engineering can be looked as an evolution or extension of traditional software engineering. It emphasizes the restricted pre-defined exchanges and correct handling of well-structured documents, the unrestricted, spontaneous and flexible exchanges of information among team members, and the concurrent control of exchanges towards the phased achievement.

After a comprehensive literature review, we provide a precise description of Cooperative Software Engineering. A process model is used to interpret how to realize the cooperative work in software development from both process viewpoint and product viewpoint. Requirements and infrastructure of tools and environments supporting cooperation are reviewed and discussed, and a typical framework and its components are described. Our future research directions are identified as follows: (1) an executable platform of cooperative software engineering and its application in large health-care information systems; (2) representation and reasoning of temporal data in cooperative work; (3) new approaches for solving resource conflicts in cooperation.

## Acknowledgement

This work is supported by the National Natural Science Foundation of China (Grant No.60373081) and the Guangdong Provincial Science and Technology Foundation (Grant No.04 5503).

## References

1. Bandinelli, S., Di Nitto, E., Fuggetta, A.: Supporting Cooperation in the SPADE-1 Environment. *IEEE Transactions on Software Engineering*, 22 (1996) 841-865
2. Altmann, J.: "Cooperative Software Development: Computer-Supported Coordination and Cooperation", PhD Thesis, Trauner, Linz. (1999)
3. Goguen, J., Lin, K.: Web-based Support for Cooperative Software Engineering. *Annual of Software Engineering*. J. C. Baltzer AG., NJ USA (2001) 167-191

4. Gaeta, M., Ritrovato, P.: Generalized Environment for Process Management in Cooperative Software Engineering. Proceedings of the 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment. (2002) 1049-1053
5. Wang, A.I.: A Process Centred Environment for Cooperative Software Engineering. Proceedings of the 14th international conference on Software engineering and knowledge engineering, (2002) 469-472
6. Bischofberger, W. R., Kofler, T., Mätzler, K.-U., Schäffer, B.: Computer Supported Cooperative Software Engineering with Beyond-Sniff. Proceedings of Software Engineering Environments, (1995) 135-143
7. Bauknecht, K., Mühlherr, T., Sauter, C., Teufel, S.: Computerunterstützung für die Gruppenarbeit, Addison-Wesley, Bonn (1995)
8. Malone, T. W., Crowston, K.: The Interdisciplinary Study of Coordination. ACM Computing Surveys, 26(1) (1994) 87-119.
9. Altmann, J., Pomberger, G.: Cooperative Software Development: Concepts, Model and Tools. Technology of Object-Oriented Languages and Systems, (1999) 194-207
10. Shi, M. L., Xiang, Y., Yang, G. X.: The Theoretics and Application of Computer Supported Cooperative Work. Publishing House of Electronics Industry, Beijing, (2000)
11. Madhavji, N. H.: The Process Cycle. Software Engineering Journal. 6(5) (1991) 234 - 242
12. Workflow Management Coalition (WfMC) and the Workflow And Reengineering International Association (WARIA), <http://www.e-workflow.org/>
13. Vessy, I., Sraavanapudi, A. P.: CASE Tools as Collaborative Support Technologies. Communications of the ACM, 38(1) (1995) 83-95
14. Altmann, J., Weinreich, R.: An Environment for Cooperative Software Development: Realization and Implications. Proceedings of the 31st Annual Hawaii International Conference on System Sciences, Collaboration Systems and Technology. IEEE, Los Alamitos (1998)
15. Weinreich, R., Altmann, J.: An Object-Oriented Infrastructure for a Cooperative Software Development Environment. Proceedings of the Fifth International Symposium on Applied Corporate Computing, ITESM, Monterrey Mexico (1997)
16. Shi, M. L., Yang, G. X., Xiang, Y., Wu, S. G.: WFMS: Workflow Management System. Chinese Journal of Computers, 22(3) (1999) 325-334
17. Li, H. C., Shi, M. L., Chen, X. X.: Business Process Modeling and Analysis in Workflow Systems. Journal of Computer Research & Development, 38(7) (2001) 798-804
18. Yu, Y., Tang, Y., Liang, L., Feng, Z. S.: Temporal Extension of Workflow Meta-model and Its Application. Proceedings of the eighth International Conference on Computer Supported Cooperative Work in Design. (2003) 293-297
19. Yu, Y., Tang, Y., Tang, N., Ye, X. P., Liang, L.: A Meta-model of Temporal Workflow and Its formalization. Proceedings of the Third International Conference on Grid and Cooperative Computing. (2004) 987-992



# An Algorithm for Cooperative Learning of Bayesian Network Structure from Data

Jiejun Huang, Heping Pan, and Youchuan Wan

School of Remote Sensing and Information Engineering,  
Wuhan University, Wuhan, 430079, P.R. China  
hjjtk@21cn.com, panhp@bigpond.com, wych@public.wh.hb.cn

**Abstract.** Bayesian network is an important and powerful method for representing and reasoning under conditions of uncertainty, and has been widely used in artificial intelligence and knowledge engineering. Structure learning is certainly the most difficult problem in Bayesian network research. In this paper we give an introduction to Bayesian networks, and review the related work on learning Bayesian networks. Then we discuss the major difficulties in structure learning, and propose an efficient algorithm for cooperative learning of Bayesian network structure from database. The experimental results from a case study prove that such an approach is feasible and robust.

## 1 Introduction

Along with the development of information technology, how to extract and utilize information resources is very important for decision-making. Bayesian network (BN), advanced by Pearl [1], is a probabilistic graphical model, which has been used for probabilistic reasoning in expert systems. Because this method has a powerful ability for reasoning and a flexible mechanism to learning, it provides an effective way to deal with incomplete data or uncertainty. Nowadays, it has been used in a number of different domains, such as medical diagnoses [2], knowledge discovery and data mining [3], image interpretation and pattern recognition [4,5]. Undoubtedly, it has been a research hotspot in artificial intelligence and knowledge engineering.

Bayesian networks also called causal probabilistic networks, belief networks, or influence diagrams [6]. The basic thinking of BN is the Bayes theorem in probability theory. It can incorporate expert knowledge and historical data for decision-making, and give a graphical representation of the domain problems and results. A Bayesian network consists of two components: one includes a set of variables and a set of directed edges between variables, which forms a directed acyclic graph (DAG); the other includes a conditional probability table (CPT), which represents the uncertainty of relationships on each variable with its parents. Suppose a data set  $\mathbf{D}$  is given, which is defined by  $n$  variables  $\mathbf{V}=\{V_1, V_2, \dots, V_n\}$ . Each variable respond to a node. Let  $\mathbf{G}$  represents a DAG;  $\mathbf{L}$  is a set of directed links;  $\mathbf{P}$  is a set of conditional probability distributions associated with every node. Then a Bayesian network BN can be noted by:

$$\text{BN} = (\mathbf{G}, \mathbf{P}) = (\mathbf{V}, \mathbf{L}, \mathbf{P}) \quad (1)$$

where

$$\mathbf{L} = \{(V_i - V_j) | V_i, V_j \in \mathbf{V}\} \quad (2)$$

$$\mathbf{P} = \{P(V_i | V_{i-1}, V_{i-2}, \dots, V_1), V_i \in \mathbf{V}\} \quad (3)$$

Using Bayes chain rule, and let  $Pa_i$  is the set of parents of the variable  $V_i$ , so we can get the joint probability distribution:

$$P(\mathbf{V}) = \prod_{i=1}^n P(V_i | Pa_i) \quad (4)$$

In the next section, we discuss the related work on learning Bayesian networks. In Section 3, we discuss the major difficulties in structure learning, and then propose an efficient algorithm for cooperative learning of Bayesian networks. In Section 5, we present a case study to prove the feasibility and effectiveness of the proposed algorithm. Eventually, Section 6 contains some remarks as well as some suggestions for future research.

## 2 Related Work

Learning Bayesian networks is currently a central area in Bayesian network research. It is a process to optimize the networks. Its goal is to find out a network model that best represents the dependent relationships of the variables in a given database. The problem can be divided into two aspects: structure learning and parametric learning. The former is to obtain the topology of the network, in other words, construct a DAG for the particular problem. The latter is to get CPT about the data set. Structure learning is the key step to perform reasoning and predicting, and is one of the important parts of the research domain. This paper focuses on structure learning.

Research on learning network models started from Chow and Liu [7]. They introduced a method for recovering simple tree-structured belief networks. Subsequently, Rebane and Pearl [8] extended Chow and Liu's method to learning polytrees-singled connected networks, so that it is able to recover exactly the polytree underlying the given data set. However, if the raw data come from a non-polytree distribution, the method is reliable. Cooper and Herskovits (1992) developed a Bayesian method for the induction of Bayesian networks from data, provided an algorithm for obtaining the most probable Bayesian network given a database of case [9]. Many other researchers have studied the Bayesian approach for structure learning [10,11]. Although this approach can handle missing data and hidden variables, it assumes a uniform distribution over all possible network structures. Lam and Bacchus [12] introduced an approach that is based on MDL principle to overcome the difficulty by avoiding the explicit definition of the structure in prior. With the MDL score, the prior of a hypothesized network structure is replaced by the description length of the structure. It is important that the length of structure is computable. Nevertheless, the MDL method still cannot obtain the best structure in the exponentially searchable space of possible structures. Singh and Valtorta [13] presented a method of recovering the structure of Bayesian networks from a database by integrating CI test based methods and Bayesian methods. Recently, more and more researchers have been conducting studies on

structure learning in the presence of incomplete data and hidden variables [14,15,16]. Many scoring criteria that are used to learn Bayesian network structures from data are score equivalent. Chickering [17] used this criterion in conjunction with a heuristic search algorithm to perform model selection or model averaging.

More and more methods and algorithms have been developed recently. Unfortunately, each of them has its disadvantages. It is just because structure learning is a complex and time-consuming work, and the difficulties mainly arise from the size of the structure space that is exponential relative to the number of variables and the acyclicity of directed links has to be guaranteed for each possible structure. In the following section, we discuss the major difficulties in structure learning and propose an algorithm that can reduce the search space and improve the efficiency of learning.

### 3 Major Difficulties in Structure Learning

The main obstacle for using Bayesian networks is to construct the domain model, that is to say, learning the structure of a Bayesian network is very hard. The major difficulties are discussed below.

#### 3.1 Discontinuity of Structure Space

The structure space refers to the set of all possible structures for a target Bayesian network to be learned from a given data set. The fact that the structure space is not continuous suggests discontinuous search strategies for enumerating possible alternative structures. This is different from parameter learning where the parameter space is continuous and thus an iterative approximating method such as EM can be used to approach the global optimum in the continuous problem space. It is possible in principle to coerce a full specification of the network including the structure and its parameters into a single vector of real-valued parameters. However, this artificial construct would have complex non-continuous derivatives. On the other hand, it is often infeasible to select a single optimal model from an exponential-sized set of models. Consequently, rather than selecting a single optimal model, an alternative approach is to look for a subset of good models.

#### 3.2 Exponential Size of Structure Space

The structure space has a size exponential related to the number of variables of the network. This renders any search-based method inefficient in finding the global optimum if the size of the network is not trivial small. For a network of  $n$  variables, let  $N_{ul}$  be the total number of all possible different undirected links,  $N_{us}$  be the total number of all possible different undirected network structures on the  $n$  variables, and then we have

$$N_{ul} = n(n-1)/2 \quad (5)$$

$$N_{us} = 2^{N_{ul}} = 2^{n(n-1)/2} \quad (6)$$

When the variables are ordered in advance so that a link can only be directed towards a variable later in the ordering, then the total of all possible directed links is

$$N_{dt} = (n-1) + (n-2) + \dots + 1 = \frac{1}{2}n(n-1) \quad (7)$$

Accordingly, the total number of all possible directed structures given a particular variable ordering is

$$N_{ds} = 2^{n(n-1)/2} \quad (8)$$

The total number of all possible different  $n$ -variable orderings is

$$N_{vo} = n! \quad (9)$$

Therefore, we can determine the number of all possible directed network structures  $N_s$  of the  $n$  variables:

$$N_s = N_{vo} \cdot (N_{ds} - 1) + 1 = n!(2^{n(n-1)/2} - 1) + 1 \quad (10)$$

### 3.3 Acyclicity Constraint

For each hypothesized structure of the network, the acyclicity constraint over directed links has to be satisfied. An acyclic directed graph will not lose its acyclicity by removing any existing directed link. Directed cycles may only appear when a new directed link is added. Generally, maintaining an ordering of the variables is an easy way to guarantee the acyclicity constraint satisfied.

### 3.4 Equivalence Class of Structures

Furthermore, not all the possible structurally different directed networks are essentially different. An important concept is the equivalence class of networks (structure) [18]. Two structures are equivalent if they exhibit equivalent functional decompositions of the full joint probability density, and therefore equivalent independence or a committee of reasonable models rather than one best model from the data.

### 3.5 Incomplete and Soft Data

Incomplete and soft data raise serious difficulties to the structure learning. Not only should hypothesizing alternative structures be motivated by statistical properties of the data, but also selecting structures and estimating parameters for a structure must rely on the data. Although the problem of incomplete soft data can be solved using EM algorithm in parameter estimation, when both structure and parameters are to be estimated, even the EM algorithm appears to be problematic.

## 4 An Algorithm for Learning Bayesian Network Structure

In this section, we propose an algorithm for cooperative learning of Bayesian network structure. Its basic thinking is to set the related parameters to reduce the search space of possible structures based on expert knowledge and prior knowledge, then to prune the fully connected potential graph through conditional independence (CI) tests. So

we can get the minimal potential graph, which is the best-undirected structure responding to the DAG, and then determine the direction of the links by the Bayesian MAP criterion. The procedure contains four steps: (1) Define the variables for a domain and prepare the data set, and then generate a fully connected potential graph; (2) In light of the expert and prior knowledge, divide the variables into two sets of variable pairs:  $\mathbf{L}_0$  is the set of variable pairs which have an undirected link and  $\mathbf{L}_1$  is the set of variable pairs which have not an undirected link. Set the maximum number of parents  $T_P$  for any variable in the underlying BN. This step may reduce the search space of possible structures and improve the efficiency of computation. (3) Use CI tests to prune the remaining links, and then obtain a minimal potential graph that approximates the undirected version of the underlying directed graph. (4) Use the Bayesian MAP criterion to determine the direction of links.

In order to clarify and illustrate the algorithm, we give the following definitions and theorem:

**Definition 1:** For a problem domain, let each node represent a variable respectively, so all of undirected links between every two nodes can construct a graphical model, called fully connected potential graph, denoted by PG.

**Definition 2:** If the random variables  $X$ ,  $Y$  and  $Z$  with a joint distribution  $P(X, Y, Z)$ , there has:  $P(X|Y, Z) = P(X|Z)$ , that is to say if the state of  $Z$  is known then no knowledge of  $Y$  will alter the probability of  $X$ , So we call the variables  $X$  and  $Y$  are Conditional independent given  $Z$  under  $P$ , denoted by  $I(X \perp Y|Z)$ .

**Definition 3:** If  $\mathbf{X}$ ,  $\mathbf{Y}$  and  $\mathbf{Z}$  are three disjoint subsets of nodes in a directed acyclic graph (DAG)  $D$ , then  $\mathbf{Z}$  is said to *d-separate*  $\mathbf{X}$  from  $\mathbf{Y}$ , denoted  $\langle \mathbf{X}|\mathbf{Z}|\mathbf{Y} \rangle_D$ , if for all paths between a node in  $\mathbf{X}$  and a node in  $\mathbf{Y}$  there is an intermediate node  $w$  such that either (1)  $w$  has converging arrows and none of  $w$  or its descendants are in  $\mathbf{Z}$ , or (2)  $w$  does not have converging arrows and  $w$  is in  $\mathbf{Z}$ .

**Theorem 1:** For any DAG  $D$  there exists a probability distribution  $P$  such that  $D$  is a perfect map of  $P$  relative to d-separation, i.e.,  $P$  embodies all the independencies portrayed in  $D$ , and no others.

The algorithm procedure can be described as follows:

### (1) Initialize the Fully Connected Potential Graph

Firstly, based on the given problem and the data case, we get the fully connected potential graph. That is, assume that there is a relationship between any of the variables, and the links are used to represent the relationships, so a fully connected PG is constructed. Written in mathematical forms as:

$$PG = (\mathbf{V}, \mathbf{L}, \Phi) \quad (11)$$

Where

$$\mathbf{V} = V_1, V_2, \dots, V_n \quad (12)$$

$$\mathbf{L} = \{(V_i - V_j) | V_i, V_j \in \mathbf{V}\} \quad (13)$$

$$\Phi = \{\emptyset(V_i, V_j), \forall (V_i, V_j) \in \mathbf{L}\} \quad (14)$$

The numbers of links  $\mathbf{L}$ :

$$|L| = n(n-1)/2 \quad (15)$$

Let  $A_X$  denotes the set of directed neighbors of  $X$ , and  $|A_X|$  denotes the number of  $X$ 's neighbors. Initialize

$$A_X = \mathbf{V} \setminus \{X\} \quad (16)$$

So  $|A_X| = n - 1$ , where  $n = |\mathbf{V}|$ .

### (2) Enter Prior Knowledge

For any of the two variables  $X$  and  $Y$ , based on the expert knowledge and prior information, set the tree parameters as:

Let  $\mathbf{L}_0 = \{(X, Y)\}$ , denote the set of variable pairs  $(X, Y)$  between which there must be an undirected link.

Let  $\mathbf{L}_1 = \{(X, Y)\}$ , denote the set of variable pairs  $(X, Y)$  between which there must not be an undirected link.

Let  $T_p$  denote the maximum number of parents for any variable in the underlying Bayesian network. It may be set by prior knowledge or commonsense knowledge which satisfied  $T_p < n-1$ , or be set  $T_p = n-1$  if no prior knowledge about this number is available.

### (3) Prune the Potential Graph

Let  $(X \perp Y | \mathbf{Z})$  denote a conditional independence test to be true,  $t_p$  denote the number of conditioning variables  $\mathbf{Z}$ . Let  $\wedge(X, Y)$  denote the minimum d-separation set of variables d-separating variables  $X, Y$ . We can prune the fully connected PG by this step, and get the minimal PG, which approximates the undirected version of the underlying directed graph. The main loop is described as bellows.

```

for (  $t_p = 0$ ;  $t_p < T_p$ ;  $t_p ++$  )
for (  $i = 0$ ;  $i \leq n$ ;  $i ++$  )
for (  $j = i + 1$ ;  $j \leq n$ ;  $j ++$  )
{
Let  $X = V_i, Y = V_j, \mathbf{U} = \mathbf{V} \setminus \{X, Y\}$ 
If (  $(X, Y) \in \mathbf{L}_0$  ), then
set  $\phi(X, Y) = \phi(X, Y) = 1$ 
else if (  $(X, Y) \in \mathbf{L}_1$  ) then
set  $\phi(X, Y) = \phi(X, Y) = 0$ 
else if  $Y \in A_X, |A_X| > t_p, |A_Y| > t_p$ 
Let  $\mathbf{z}$  enumerate the subsets of size  $t_p$  of  $(A_X \cup A_Y) \setminus \{X, Y\}$ 
if  $(X \perp Y | \mathbf{z})$  then
cut the link between  $X, Y$ :  $\phi(X, Y) = 0$ ,
 $A_X = A_X \setminus \{Y\}, A_Y = A_Y \setminus \{X\}$ 
set the d-separation set  $\wedge(X, Y) = \mathbf{z}$ 
else set  $\phi(X, Y) = I(X; Y | \mathbf{Z})$ 
}

```

In the process: (1) If  $|A_X|=0$  at a certain time, then the variable  $X$  is considered isolated from all the other variables, and it will not be included in the future steps. In this way, isolated variables can also be handled. (2) It is because that  $t_p$  denotes the number of conditioning variables, starts from zero and increases one by one, the d-separation set  $\wedge(X, Y)$  obtained for any two variables  $X, Y$  must be a minimal d-separation set not including irrelevant variables.

#### (4) Edges Direction

According to Cooper and Herskovits [9], we can find the most probable network structure given the database by

$$P(\mathbf{D} | S) = \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \quad (17)$$

In the light of the variables in the domain, we can get the direction of some edges based on prior knowledge. Let  $(X-Y)$  denote that there is an undirected link between  $X$  and  $Y$ ,  $(X \perp Y)$  denote that  $X$  and  $Y$  are not adjacent, and  $(X \rightarrow Y)$  denote there is a directed link from  $X$  to  $Y$ . And let  $X \rightsquigarrow Y$  denote that there is a directed path from  $X$  to  $Y$ . So we can get the direction of links with the algorithm by following steps in the given order:

1. Enter prior knowledge: If the directed links are provided by the user, add the direction as they provide.

2. Rule1: Resolving the converging arrows.

For  $(X, Y, Z)$  enumerate all triples of variables:

If  $(X-Z) \& (Y-Z) \& (X \perp Y) \& Z \notin \wedge(X, Y)$  then orient  $X-Z-Y$  as  $X \rightarrow Z \leftarrow Y$ .

3. Rule2: Resolving transitive arrows.

For  $(X, Y, Z)$  enumerate all triples of variables that do not form converging arrows  $(X \rightarrow Z \leftarrow Y)$ :

If  $(X \rightarrow Z) \& (Y-Z) \& (X \perp Y)$  then orient  $Z-Y$  as  $Z \rightarrow Y$ .

4. Rule3: Resolving acyclic arrows.

For  $(X, Y)$  enumerate all undirected links that remain to be oriented:

If  $(X \rightsquigarrow Y) \& (X-Y)$ , then orient  $X-Y$  as  $X \rightarrow Y$ .

5. Repeat step 2 and 3 until no more edges can be oriented.

6. If there still have edges undirected, use the Bayesian MAP criterion to determine the direction. Use formula (17) to compare the aposterior probability of the model given the data.

if  $P(S_{X \rightarrow Y} | D) > P(S_{Y \rightarrow X} | D)$

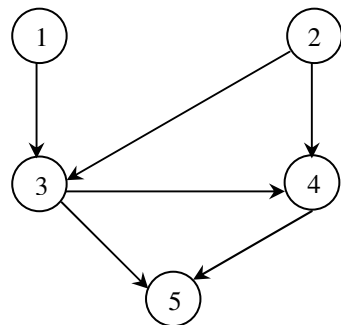
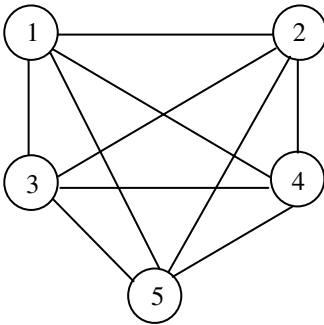
then the orient  $X-Y$  as  $X \rightarrow Y$ .

Where  $S_{X \rightarrow Y}$  and  $S_{Y \rightarrow X}$  denote the structure with the direction between  $X$  and  $Y$  respectively. And  $D$  denotes the given data set.

In this phase, we can determine all the direction of the edges, and finally obtain a directed acyclic graph (network structure).

## 5 A Case Study

Bayesian network has a powerful ability to utilize and represent the information or knowledge, so it has been extensively used to business and commerce [19]. Based on the algorithm, we show a case study for market modeling. In the product-marketing database, there are 5 variables, and each of them has two states, described as packaging (yes, no), customer service (yes, no), price (elevated, unchanged), total of selling (increased, reduced), revenues (increased, reduced). The data set contains 1500 records. In the light of the variables in the domain, we first get the fully connected potential graph (PG) of market model (as shown in Fig. 1). Using the algorithm provided in Section 4, a Bayesian network of market model can be constructed (as shown in Fig. 2). The model can help us make decisions in business implementation and optimize selling profitability. Furthermore, it can guide the businessman to take more precise marketing actions and improve the competitive advantage by marketing more intelligently.



**Fig. 1.** Fully Connected PG of Market Model    **Fig. 2.** Bayesian Network of Market Model

## 6 Conclusion and Future Work

In recent years, many researchers pay a great attention in learning Bayesian networks from data. Structure learning is the hotspot in the research domain. How to improve the efficiency and accuracy of learning has been the main theme in this area. In this paper, we propose an algorithm for cooperative learning of Bayesian network structure. The basic idea is to set the related parameters to reduce the search space of possible structures based on expert knowledge and prior knowledge, then to prune the fully connected potential graph through conditional independence (CI) tests. Using conditional independence (CI) tests, we can prune a fully connected potential graph to a best PG, which is expected to approximate the undirected version of the underlying directed graph. Eventually, the Bayesian MAP criterion is used to determine the direction of links. A case study in business intelligence proved the feasibility and effectiveness of the proposed algorithm. Our future work includes the feasibility study of applying the proposed algorithm to complex problems or very large potential graphs. Meanwhile, the sensitivity for the database size and the usability for incomplete data are also considered as our further research directions.



## Acknowledgement

The work presented in this paper was supported by the National Natural Science Foundation of P. R. China (No. 60175022).

## References

1. Pearl, J.: Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29(3) (1986) 241-288
2. Beinlich, I. A., Suermondt, H. J., Chavez, R. M., et al.: The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks. *Proceedings of the European Conference on Artificial Intelligence in Medicine*, (1989) 247-256
3. Heckerman, D.: Bayesian Network for data mining. *Data mining and knowledge discovery*, 1 (1997) 79-119
4. Kumar, V. P., Desai, U. B.: Image interpretation using Bayesian networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1) (1996) 74-77
5. Piater, J.H., Grupen, R.A.: Feature learning for recognition with Bayesian networks, *The 15th International Conference on Pattern Recognition*, 1 (2000) 17-20
6. Heping, Pan., Lin, Liu.: Fuzzy Bayesian networks - a general formalism for representation, inference and learning with hybrid Bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 14(7) (2000) 941-962
7. Chow, C. K., Liu, C. N.: Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3) (1968) 462-467
8. Rebane, G., Pearl, J.: The recovery of causal poly-trees from statistical data. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, (1987) 222-228
9. Cooper, G., Herskovits, E.: A Bayesian method for the induction of Bayesian networks from data. *Machine Learning*, 9 (1992) 309-347
10. Buntine, W.: Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, 2 (1994) 159-225
11. Heckerman, D., Geiger, D., Chickering, M. D.: Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 1995, 20(2) (1995) 197-243
12. Lam, W., Bacchus, F.: Learning Bayesian Belief Networks: An approach based on the MDL Principle. *Computational Intelligence*, 10 (1994) 269-293
13. Singh, M., Valtorta, M.: Construction of Bayesian network structures from data: A brief survey and an efficient algorithm. *International Journal of Approximate Reasoning*, 12 (1995) 111-131
14. Chickering, D. M., Heckerman, D.: Efficient approximations for the marginal likelihood of incomplete data given a Bayesian networks. *Machine Learning*, 29 (1997) 181-212
15. Kwoh, C. K., Gillies, D. F.: Using hidden nodes in Bayesian networks. *Artificial Intelligence*, 88(12) (1997) 1-38
16. Nikovski, D.: Constructing Bayesian networks for medical diagnosis from incomplete and partially correct statistics. *IEEE Transactions on Knowledge and Data Engineering*, 12 (2000) 509-516
17. Chickering, D. M.: Learning equivalence classes of Bayesian network structures. *Journal of Machine Learning Research*, 2 (2002) 445-498
18. Chickering, D.: Learning equivalence classes of Bayesian network structures. In: *Proc. of Twelfth Conference on Uncertainty in Artificial Intelligence*, (1996) 150-157
19. Kozłowski, D. C., Yamamoto, C. H., Carvalho, F. L.: Using Bayesian networks to build data mining applications for an electronic commerce environment. *2002 IEEE International Conference on Systems, Man and Cybernetics*, (2002) 72-77

# Non-violative User Profiling Approach for Website Design Improvement

Jiu Jun Chen<sup>1,2</sup>, Ji Gao<sup>1</sup>, and Song En Sheng<sup>2</sup>

<sup>1</sup> College of Computer Science, Zhejiang University,  
Hangzhou 310027, Zhejiang, China

<sup>2</sup> Zhejiang University of Technology,  
Hangzhou 310014, Zhejiang, China  
rackycjj@zju.edu.cn

**Abstract.** Web user profiles are widely applied to complete the high-quality website design and the personalized web services. One issue in this technology is the study of contradiction between the collection of personal data and the protection of user privacy. In order to alleviate the contradiction, this paper provides a new profiling approach, namely non-violative user profiling, which integrates the non-violative strategy and Markov user model ideas. It can extract more exact information automatically with the participation of web users. It has been implemented in our experimental website to identify the design defects and validate the effectiveness of the proposed approach.

## 1 Introduction

Web user profiling technology [1] can extract the user information to acquire the most complete picture of the web visitor. The information, so-called user profiles can help in improving the design of website and customizing the content to the needs of specific users [2].

Bygrave [3] defined the user profiling as “the process of inferring a set of characteristics about an individual person or collective entity and then treating that person or entity in the light of these characteristics”. In the Internet application, user profiling is the process of gathering information specific to each web user, either explicitly or implicitly, which refers to a set of user preferences or settings including his or her interests and navigational behaviors.

User profiling can be classified into explicit user profiling and implicit user profiling. Explicit user profiling is the process of analyzing explicitly users’ static and predictable information which usually comes from electronic registration or survey forms, to achieve the characters of the user. The advantages of this approach include:

- users interact with the system directly and have choice to agree or reject the process of profiling;
- users can easily adapt the profiles elements about themselves;
- the system can make some assumptions about a user in advance, and improve the profiling process.

Explicit user profiling also has a number of obvious disadvantages:

- it is difficult to ensure that all users will voluntarily provide information to the system;
- answers provided by users may not reflect their own inspiration accurately;
- answers of some users may prone to their own subjectivity;
- the profile is static and not flexible enough to take a user's interest changes into account.

However, explicit user profiling also has some disadvantages:

- users maybe put off when realizing their individual information is collected;
- user profiles may not be fully certain;
- general interests of the user cannot be traced at real time.

Unlike explicit user profiling, implicit user profiling is the process of analyzing implicitly a user's activities, e.g., time spent, present position, and navigational history to determine what the user is interested in. It can be processed automatically, and the profiles can be updated and adjusted continuously.

Collecting and understanding of personal data is the key of the profiling process. So the risk of privacy for users is raised correspondingly, just as introduced in [4] that personal details are often used and sold without user's consent. That is the reason that web users are very concerned about threats to their personal information, and extremely worried about divulging individual data as well as being tracked. It is reported in [5] that most of web users counteract by leaving web sites that required registration information or having entered fake registration information.

But it also reported that eighty-one percent of web users want web sits to ask for permission to user personal data [5], and more than thirty percent of them are willing to give out personal data for getting something valuable in return [6].

So the study of the contradictive situation is necessary. Some previous works have been done. In [7], the author believed that improving the users' control over their owner data while profiling could improve the protection of user privacy. Dickinson et al. [8] proposed to use policy-based access control to improve the user's control over profiling. A solution, so-called ePerson, is also suggested, in which the information about the individual data is directly under the user's control. In [9][10], a new breed of technologies, namely Privacy-Enhancing Technologies, have been developed to help individual users take control over their data being collected. Pearson [11][12] used trusted agents that exploit Trusted Computing Platform technology to build up one or more profiles corresponding to users.

Based on those works, the following problems should be studied further:

- how to define the user's role while profiling;
- how to construct an effective dynamic user model to realize data mining and information update;
- how to reuse the profiles.

In this paper, we will address these problems through the following steps:

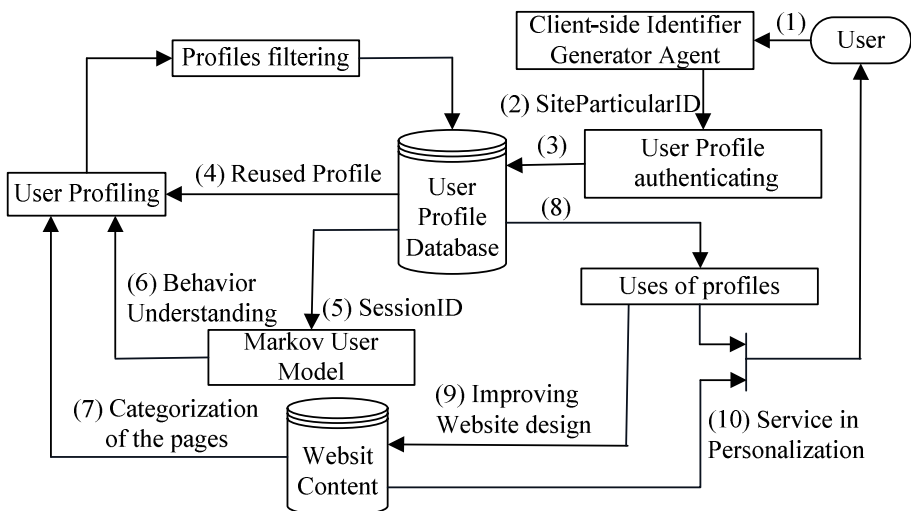
- with the use of a non-violative strategy, user privacy can be protected while profiling;

- a stochastic user model is defined to mine the users' behavior automatically;
- profiles are modified or filtered with user's participation;
- as an example, the solution has been implemented to improve the design of the website.

The rest of the paper is organized as follows: in the next section, the non-violative user profiling model is presented in brief. The non-violative strategy, as the key criterion in our profiling process, is studied. And then the stochastic user model is constructed, and the algorithms of the model are designed. Finally, we implement the method in our website, and analyze the results obtained.

## 2 Non-violative Web User Profiling

Our approach, namely, non-violative user profiling, integrates the non-violative strategy and Markov user model ideas. The strategy protects the personal privacy, and the Markov user model learns the characters of the web user automatically.



**Fig. 1.** Non-violative user profiling model

As shown in Figure 1, when a web user browses the website, he or she will be assigned with two identifiers. One is Site Particular ID, which is generated by Client-side Identifier Generator Agent. The other is Session ID, which is generated by the User Profile Database. The former is the true identifier, and the latter is a random identifier for visiting. “Double identifier” achieves the idea of non-violative strategy. Then a Markov user model based on “Double identifier” collects and extracts the user’s navigational behaviors to acquire user profiles together with the reused profiles, website structure and page contents. The extracted user profiles will be in direct control under the user who can modify or filter the profiles on his own behalf. The profiles are then stored in User Profile Database with the user’s acknowledgment. The

profiles in Database are then used in applications, such as website design and personalized web services.

## 2.1 Non-violative Strategy

Non-violative strategy is viewed as an enhanced anonymous mode to protect the user's privacy while profiling. The user has right to control his or her individual data, and decides which elements of the user profile are revealed and what purposes for later use of these data are. They can inspect, block, rectify and erase both the data that they themselves provided, and specifically the assumptions that the system inferred about them.

Anonymous model works as the release of the user's true identity to a pseudo-identity to protect the interests of the user while data are collected. Web users that remain anonymous are more likely to provide data about themselves [13]. Because the user may have multiple separate identities, which may or may not reference the same user profile [14], the system will not be able to recognize the user returning, and so some information about the user may not be re-used.

In order to address this dilemma, the user identifier can be global in scope [15] and persistent over a long period. But this method affords the user little in privacy protection, since the same identifier is given to each site, allowing long term tracking of the user between different sites. In general, there is no need for the identifier to be global and persistent over time. In [8], Dickinson et al. present a solution that uses a sit scope identifier instead of a single, global identity.

“Double Identifier” achieves the idea of non-violative strategy.

One identifier of the web user, namely “site-particular identifier” generated by a client-side identifier generator agent, is used to identify special user for each visited website. Different website has different sit particular identifier, which would prevent the user's navigational behavior being tracked between sites and protect the user's privacy.

The other identifier is allocated by the website profile store. It is a random session identifier which is used for identifying the user's within-site behavior. Because the session identifier changes each session, it cannot be used by the website to track the user over a long period.

We believe that it is sufficient that the identifier in term of the profile can be mapped back to a specific user. So the mapping relationship between site-particular identifier and session identifier is only managed by the user profile database. It is defined as the following statement:

*User ProfileDB* < *Profile<sub>i</sub>*, *SessionIDGenerator()*, *ID Recognise()*, *IsIDMapping()* >

It includes three functions:

- (a) constructing the mapping relationship between site-particular identifier and session identifier by *IsIDMapping(SessionID<sub>k</sub>, SiteParticularID<sub>j</sub>)* function and continuously updating and adjusting the user profiles;
- (b) recognizing the user returning and reusing user profiles *SitParticularID<sub>j</sub>* will map back to *Profile<sub>i</sub>* by function *ID Recognise(Profile<sub>i</sub>, SitParticularID<sub>j</sub>)*;

(c) generating a random session identifier  $SessionID_k$  for each visiting.

We define Client-side Identifier Generator Agent ( $IDGeneratorAgent$ ) as the following:  $IDGeneratorAgent \langle User_i, WebSite_j, SiteParticularID_j \rangle$ . It generates  $SiteParticularID_j$  for  $User_i$  while visiting  $WebSite_j$ .

## 2.2 Mathematical Modeling

Based on non-violative strategy, a stochastic user model, namely, Markov user model is defined to extract and represent user's within-site behaviors. We model user's navigational activities as a Markov process for the following reasons: firstly, the information itself may change; secondly, the web user is largely unknown from the start, and may change during the exploration.

We define some terms associated with the model:

- *Definition 1: State.* A state is defined as a collection of one or more pages of the website with similar functions. In our model, state contains other two special states: Entry and Exit. We assume that web user always enters the Entry state before making any request, and resides in Exit state after exiting the process;
- *Definition 2: Transition.* A transition occurs with the request for one page that belongs to state  $j$  while the user resides in one of the pages belonging to state  $i$ ;
- *Definition 3: Transition probability matrix ( $P$ ).* It is a stochastic matrix with elements  $p_{ij}$ , where  $p_{ij}$  is the probability of transition from state  $i$  to state  $j$ . This matrix can capture a compact picture of users' navigational behavior;
- *Definition 4: Holding time ( $t_{ij}$ ).* It is the time which the process remains in one state before making a transition to another state;
- *Definition 5: Mean holding time matrix ( $\bar{T}$ ).* It is a matrix with element  $\bar{t}_{ij}$ , where  $\bar{t}_{ij}$  is the mean of  $t_{ij}$ .

We use transition probability matrix and mean holding time to describe the web user behavior. A set of four elements defines a discrete Markov model:  $\langle SessionID_k, P, \bar{T} \rangle$ , where:

- $SessionID_k$  is a random session identifier of  $User_i$ , which is used for identifying the with-site user behavior;
  - $P$  is a transition probabilities matrix, which is of size  $(N+2) \times (N+2)$ ;
  - $\bar{T}$  is a mean holding time matrix. They can be found out by the algorithm:
- (i) For the first request for state  $s$  in the session, add a transition from Entry state to the state  $s$  and increment  $TranstionCount_{1,s}$  in a matrix  $TranstionCount[i, j]$  by 1, where  $TranstionCount[i, j]$  is a matrix to store the transition counts from state  $i$  to state  $j$ ;

- (ii) For the rest of user' requests in the session, increment the corresponding transition count of  $TransitionCount_{i,j}$  in the matrix, where  $i$  is the previous state and  $j$  is the current state;
- (iii) For the last page request in the session, if the state is not the explicit exit state then add a transition from the state to exit state and increment  $TransitionCount_{s,(n+2)}$  value by 1;
- (iv) To find out the time spent in state  $i$  before the transition is made to state  $j$  for any transition from state  $i$  to state  $j$ , except the transition from entry state, and the transition to the exit state. If this time belongs to the interval  $k$  then, increment  $HoldTimeCount_{i,j,k}$  by 1 in a three-dimensional matrix  $HoldTimeCount[i, j, m]$ , where,  $HoldTimeCount_{i,j,k}$  is the number of times the holding time is in the interval  $k$  at state  $i$  before the transition is made to state  $j$ .
- (v) Divide the row elements in matrix  $TransitionCount[i, k]$  by the row total to generate transition probability matrix  $P$ , whose element is  $p_{i,j}$ :

$$p_{i,j} = \frac{TransitionCount_{i,j}}{\sum_k TransitionCount_{i,k}} \quad (1)$$

- (vi) Find out the interval total for each transition from state  $i$  to state  $j$  in  $HoldTimeCount[i, j, m]$ . Divide frequency count in each interval with the interval total to find out the probability of occurrence of the corresponding intervals. Repeat this to generate  $HoldTimeProbability[i, j, m]$ . whose element defined as follows:

$$HoldTimeProbability_{i,j,m} = \frac{HoldTimeCount_{i,j,m}}{\sum_n HoldTimeCount_{i,j,n}} \quad (2)$$

- (vii) Multiply each interval with the corresponding probability to generate mean hold times ( $\bar{t}_{ij}$ ), which is the elements of matrix  $\bar{T}$ .

$$\bar{t}_{ij} = \sum_m m \times HoldTimeProbability_{i,j,m} \quad (3)$$

### 2.3 Profiles Modification

Markov user model can basically represent "user profile" of within-site behavior. Taking the advantage of other information such as user input data, web content and structure, we can infer in a set of assumptions about the user, e.g., interests, characters and preferences.

We define user profiles as a set of selected characteristics for a special user or a group with similar interests. The representation would be as follows, where,

$$User\ Profile_i = \langle SessionID_k, \sum_j^{valid} Char_{ij}, \sum_l^{invalid} Char_{il} \rangle \quad (4)$$

$SessionID_k$  is a random session identifier of  $User_i$ , which can be mapping back to the user's real identifier by User profiling database.

$Char_{ij}$  is the characters associated with  $SessionID_k$ , including the information that the user offer himself and the assumption that the system infer from his navigational behaviors.

$\sum_j^{valid} Char_{ij}$  corresponds to the user profiles with the user's acknowledge. It is the sub-set of  $Char_{ij}$  and the part of the user profiles that can be used for some purposes.

$\sum_l^{invalid} Char_{il}$  is the a sub-set of characters that the user does not agree. This information will not be used in any manner.

### 3 Implementation and Example

In this session, we use non-violative user profiling approach in an example website to analyze the user navigational behaviors for improving the website design. The example site is a used-goods exchanged platform. Based on the functions of web pages, they were categorized into 12 states in advance:

- (1) Entry;
- (2) Home;
- (3) Login;
- (4) Register;
- (5) Goods Search;
- (6) Seller Data;
- (7) Leave word;
- (8) Goods List;
- (9) Goods Register;
- (10) Customer Support;
- (11) About Us;
- (12) Exit.

Under the non-violative strategy, we used Markov model to collect and understand the user's behaviors in our website. We got the information about the transition probability matrix and mean holding time matrix, which represented in Table 1. Based on the information, some design defects of the website can be identified and fixrd in order to improve the performance of the website:

- (a) Some states, such as *Login* (0.36), *Register* (0.48), *Goods search* (0.42), *Goods list* (0.34), and *Goods Register* (0.58) have high probability values associated with self-loops.

High probability values associated with the *Register* and *Login* might be the result of a visitor repeatedly registering in many names or logging in a new account from the present login. A more reasonable cause is the "intermediate



page” design problem after reviewing of the site design. We found that both *Login* and *Register* contain one intermediate page that used to show the link of *Seller Register (or Login) pages* and *Customer Register (or Login) pages*, and the user unnecessarily make two process while registering. Intermediate page can be replaced by dropdown menus to solve this problem.

The transition from *Goods search* to itself indicates the user repeated search for some goods. It is a good sign for our website. A similar explanation can be used to *Goods list* state and *Goods Register* state.

**Table 1.** Transition Probability Matrix and Mean Holding Time Matirx. The first column in the table is not displayed, whose values are all zero because of the supposition that no transition can be made to Entry state from any state. Two rows are associated with each state. The upper rows are the transition probability values and the lower rows are the mean holding times in minutes. The first state, namely entry state, owns one row, which corresponds to the transition probability values and its mean holding times are null because it is virtual state.

State	2	3	4	5	6	7	8	9	10	11	12
1	0.30	0.13	0.01	0.11			0.35		0.10		
2	0.01	0.21	0.02	0.05			0.23		0.01	0.06	0.43
	0.52	0.86	0.80	2.27			1.20		0.61	1.50	
3		0.36	0.02	0.02	0.10	0.14	0.04	0.22			0.12
		0.54	0.52	0.50	2.36	2.52	1.23	0.55			
4	0.06	0.28	0.48								0.18
	0.5	0.91	0.50								
5				0.42	0.20	0.18	0.10				0.10
				0.53	0.50	2.78	0.68				
6		0.12		0.14	0.05	0.20			0.12		0.37
		2.10		0.50	2.50	0.62			1.85		
7	0.02	0.10		0.45	0.20				0.23		
	3.12	2.78		1.25	0.67				1.32		
8				0.12	0.32		0.34		0.11		0.11
				0.87	0.50		0.50		3.36		
9		0.04			0.33			0.58			0.05
		2.64			0.61			0.55			
10	0.01	0.06	0.01	0.08	0.02	0.06	0.05		0.42	0.05	0.24
	5.60	0.85	0.52	1.37	0.77	1.52	1.56		1.78	2.53	
11	0.21	0.06		0.28			0.23			0.12	0.10
	2.52	7.20		1.21			1.18			2.66	
12											1.00

- (b) Some transitions having high probability can be explain reasonably. For example, Transition from *Goods Register* state to *Goods & Seller* detail state has a high probability ( $p_{9 \rightarrow 6} = 0.33$ ) only because some sellers may want to check the information after they register their goods message. But the high probability ( $p_{2 \rightarrow 12} = 0.43$ ) from *Home* state to *Exit* state may alarm as it may indicate that the user leave the site immediately after they go through the *Home* page. After design review, we found that the site brings all the uses who log out formally from the site to *Home* page. So the probability is high.
- (c) Checking the state transitions, we can find some “site navigability” problem. We found that no user exited from the states *Leave word to seller*

(  $p_{7 \rightarrow 12} = 0$  ). A design review indicates that the site does not permit the user to exit. It takes the user back to the *Goods search* state. It is evident that most users may check their information that leaved to the seller. So the direct link from *Leave word to seller* state to *Good & Seller detail* state is necessary that will increase site navigability.

- (d) As shown in Table 1, most of the mean holding time is small except the case from *About us* state to *Login* state (  $\bar{t}_{11 \rightarrow 3} = 7.2$  ). This transition probability is very low (  $p_{11 \rightarrow 3} = 0.06$  ), so it can be viewed as an occasional case. It means it is an occasional transition from *About us* state to *Login* state.

## 4 Conclusions

This paper provides a new profiling method, namely, non-violative user profiling to help improve the website design. The main technical contribution is the non-violative strategy, user profile reuse, a Markov user model and the algorithm to implement the model. Non-violative strategy views the web users as the most important participants, which alleviate the contradiction between the collection of personal data and user privacy concern. A Markov model is constructed to understand the user's behavior automatically and "double identifier" method is used to solve the profile reuse problem. The non-violative user profiling method is implemented in an example site to identify the design defects of website. As future work, we will improve our approach to an open and extensible component service for more uses with more privacy protection.

## References

1. Chan, P.: A non-invasive learning approach to building web user profiles, Workshop on Web usage analysis and user profiling. Fifth International Conference on Knowledge Discovery and Data Mining, San Diego, (1999) 7-12.
2. Eirinaki, M., Vazirgiannis, M.: Web mining for web personalization. ACM Transactions on Internet Technology, 3(1) (2003) 1-27.
3. Bygrave, L.: Electronic Agents and Privacy: A Cyberspace Odyssey. International Journal of Law and Information Technology, Oxford University Press, 2001, 9(3), pp. 275-294.
4. Scribber, K.: Privacy@net - An International comparative study of consumer privacy on the Internet. Consumers International, (2001).
5. Fox, S.: Trust and Privacy Online, Why Americans Want to Rewrite the Rules. The Pew Internet & American Life Project, (2000).
6. Personalization & Privacy Survey, Personalization Consortium (2000), <http://www.personalization.org/Survey>
7. Casassa, M. M., Pearson, S., Bramhall, P.: Towards Accountable Management of Identity and Privacy: Sticky Policies and Enforceable Tracing Services, HPL-2003-49 (2003).
8. Dickinson, I., Reynolds, D., Banks, D., Cayzer, S., Poorvi, V.: User profiling with privacy: a framework for adaptive information agents, Intelligent Information Agents. LNAI 2586, Springer-Verlag Berlin Heidelberg, (2003) 123-151.

9. Clarke, I., Sandberg, O., Wiley, B. Hong, T.: Freenet: A distributed anonymous information storage and retrieval system. *Proceedings of the Workshop on Design Issues in Anonymity and Unobservability, Berkeley, (2001)* 311-320.
10. Senicar, V., Jerman, B. B., Klobucar, T.: Privacy-enhancing technologies: approaches and development. *Computer Standards & Interfaces, 25(2) (2003)* 147-158.
11. Pearson, S.: A Trusted Method for Self-profiling in e-Commerce. *LNAI 2631, Springer-Verlag Berlin Heidelberg, (2003)* 177-193.
12. Pearson, S.: Trusted Agents that Enhance User Privacy by Self-Profiling. *Proceedings of AAMAS Workshop Special track on privacy, Bologna, (2002)* 113-121.
13. Cranor, L. F., Reagle, J., Ackerman, M. S.: Beyond Concern: Understanding Net Users' Attitudes About Online Privacy, TR 99.4.3, AT&T Labs Research, (1999).
14. Bowbrick, S.: Multiple Identities in the Online World. *Proceedings of First DI Workshop, hyperion.co.uk, (2000)*.
15. Stuart, G. S., Paul, F. S.: Authentic Attributes with Fine-Grained Anonymity Protection, *Financial Cryptography 2000, LNCS1962, Springer-Verlag Berlin Heidelberg, (2000)* 276-294.

# Generative Design in an Agent Based Collaborative Design System

Hong Liu, Liping Gao, and Xiyu Liu

School of Information and Management, Shandong Normal University,  
Jinan, 250014, P.R. China

lhsdcn@jnn-public.sd.cninfo.net

**Abstract.** In this paper, we present a generative design method for supporting creative conceptual design in a multi-agent system. The hierarchical multi-agent system architecture offers a promising framework and an evolutionary computational approach provides a basis for supporting generative design in distributed environments. The evolutionary approach relies upon a tree structure rather than a known binary string in general GA (Genetic Algorithm). The system can generate simple 2D sketch shapes, which are generated by using general mathematical expressions, and selected shapes are dealt with to form 3D components. These components are classified and saved in a SQL sever based database (component base). The complex design is implemented via combination of different components that come from the component base. The layout of components is generated by a genetic algorithm. The generative process is illustrated through an architectural design example.

## 1 Introduction

Conceptual design is essentially a creation process. It is the creation of functions to fulfill customer needs, and the creation of forms and behaviors to realize those functions. Early-stage design ideas have a large impact on the cost and quality of a product. Designers have the freedom to generate and explore ideas without being constrained by parameters that exist at the later design stages. If many ideas are created during conceptual design, there can be plenty of options to choose from, and consequently it is more likely that a good design can be attained.

Creativity plays a central role in conceptual design. It is associated with a cognitive process that generates a design solution, which is novel or unusual and satisfies certain requirements. Creativity is a subjective activity that depends on the designer's dower, background, personality, and environment that stimulates his/her inspiration.

Conventional design software does not take human cognition into account. In recent years there has been an increasing development of computer-aided design tools to support the design process in areas. Although many breakthroughs occurred in the development of CAD systems during the last two decades, there remains a large limitation in supporting creativity in conceptual design phase. A large part of conceptual design activity still depends largely on the creative abilities of the human designers.

Generative Design is an excellent snapshot of the creative process from conceptual framework through to specific production techniques and methods. It is ideal for aspiring designers and artists working in the field of computational media, especially

those who are interested in the potential of generative/algorithmic/combinational/emergent/ visual methods and the exploration of active images.

In this paper, we present a generative design method for supporting creative conceptual design in a multi-agent system. The hierarchical multi-agent system architecture offers a promising framework for dynamically creating and managing design tasks in distributed design environment. An evolutionary computational approach that relies upon a tree structure rather than a known binary string in general GA (Genetic Algorithm) is used in this system. The system can generate simple 2D sketch shapes, which are generated by using general mathematical expressions, and selected shapes are dealt with to form 3D components. These components are classified and saved in a SQL sever based component base. The complex design is implemented by combining different components that come from the component base.

The remainder of this paper is organized as follows. Section 2 reviews related work for computational models of multi-agent system and generative design. Section 3 introduces the framework of a multi-agent system. Section 4 presents the design agent and the tree structure based genetic algorithm. Section 5 describes the assemble model of components. In Section 6, an architectural design example is presented for showing how to use the tree structure based genetic algorithm and mathematical expressions to generate 2D sketch shapes and 3D images. Section 7 summarizes the paper and gives an outlook for the future work.

## 2 Related Work

### 2.1 Multi-agent System for Creative Design

Designing is an activity during which the designers perform actions in order to change the environment. By observing and interpreting the results of their actions, they then decide on new actions to be executed on the environment. This means that the designer's concepts may change according to what they are "seeing", which itself is a function of what they have done. We may speak of a recursive process, an "interaction of making and seeing" [1]. This interaction between the designer and the environment strongly determines the course of designing. This idea is called situatedness, whose foundational concepts go back to the work of Dewey [2] and Bartlett [3]. Situatedness provided the basis for a new approach to developing computational models of individual creative accounted for the interaction of the process with its environment [4]. This produced the opportunity to develop novel approaches to way individual creative agents interact with each other in terms of creativity [5].

The agent does not simply react reflexively in its environment but uses its interpretation of its current environment and its knowledge to produce a response [6]. Situatedness has a particular explanatory power in the area of design research, as designing has been recognized as an activity that changes the world in which it takes place [7]. Experimental studies [8, 9] have characterized designing as an interaction of the designer with their environment: after changing the environment (e.g. by means of sketching), the design agent observes the resulting changes (i.e. the sketch) and then decides on new (sketching) actions to be executed on the environment. This means that the agent's concepts may change according to what it is "seeing", which itself is a

function of what it has done. As a consequence the agent can be exposed to different environments and produce appropriate responses. A framework for situated cognition in a design agent has been developed by Gero and Fujii [10]. Based on this work, a number of design agents that embody situatedness have recently been implemented [11, 12].

We can view situatedness as the ability of an agent to construct external as well as internal representations as a function of the current situation, which is the agent's interpreted environment. In other words, a situated agent can be exposed to different environments and produce appropriate responses [13]. The agent's knowledge is thus grounded in its interactions with the environment rather than predefined and encoded by a third party.

## 2.2 Generate-and-Test Model of Creative design

Recognizing the need for a unified model for supporting creative design in a multi-agent system, Liu [14] presented a synthesis of the personal and socio-cultural views of creativity in a single model. Liu realized that the existing models of personal creativity complemented the socio-cultural models by providing details about the inner workings of the creative individual missing from the models of the larger creative system.

Liu proposed a dual generate-and-test model of creativity as a synthesis of Simon et al's model of creative thinking [15] and Csikszentmihalyi's systems view [16]. As its name suggests, the dual generate-and-test model of creativity encapsulates two generate-and-test loops: one at the level of the individual and the other at the level of society.

The model unifies Simon et al's and Csikszentmihalyi's models of creativity to form a computational model of creativity that shows how personal and socio-cultural views of creativity can be modeled in a single system. The model shows that it is possible to cast Csikszentmihalyi's systems model in computational terms and thereby provides us with a useful framework for developing a multi-agent system [17].

## 2.3 Generative Design

Generative design describes a broad class of design where the design instances are created automatically from a high-level specification. Most often, the underlying mechanisms for generating the design instances in some way model biological processes: evolutionary genetics, cellular growth, etc. These artificial simulations of life processes provide a good conceptual basis for designing products.

Evolving design concepts by mutating computer models in a simulated environment is now a well-established technique in fields as diverse as aeronautics, yacht design, architecture, textile design, fine art and music [18]. Some of the work was performed by Professor John Frazer, who spent many years developing evolutionary architecture systems with his students. He showed how evolution could generate many surprising and inspirational architectural forms, and how novel and useful structures could be evolved [18, 19]. In Australia, the work of Professor John Gero and his colleagues also investigated the use of evolution to generate new architectural forms.

This work concentrates on the use of evolution of new floor plans for buildings, showing over many years of research how explorative evolution can create novel floor plans that satisfy many fuzzy constraints and objectives [20]. Professor Celestino Soddu of Italy uses evolution to generate castles and three-dimensional Picasso sculptures [21].

However, the development of evolutionary design tools is still at its early stage. So far, many genetic algorithms have been used and tested only in design problem solution with small scope. The research and development of design support tools using evolutionary computing technology are still in process and have huge potential for the development of new design technology.

### 3 Multi-agent System Architecture

Multi-agent design system is concerned with how a group of intelligent agents can cooperate to jointly solve problems. Design is a complex knowledge discovery process in which information and knowledge of diverse sources are processed simultaneously by a team of designers involved in the life phases of a product. Complex design generally combines automated software components with human decision-makers, making it imperative to provide support for both human and computational participants in the design process [22].

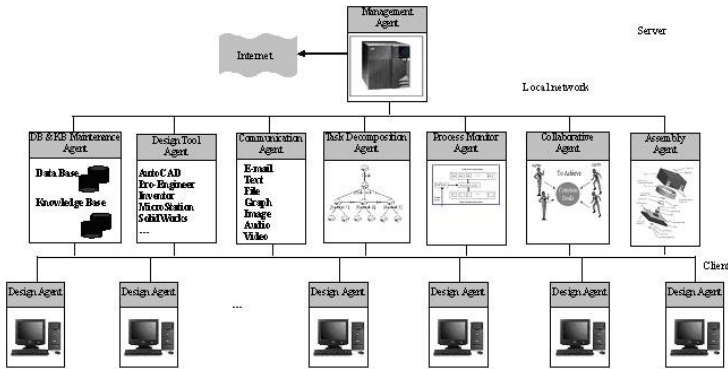


Fig. 1. The general architecture of a multi-agent design system

A multi-agent architecture has been developed for the integration of individual creative design activities. This is based on cooperating intelligent entities in the sub-domains which make decisions through negotiation, using domain-specific knowledge both distributed among the entities and accessible to them. Using this architectural framework, an agent-based system has been developed for creative conceptual design and dynamic process management.

The general architecture of a multi-agent design system is organized as a population of asynchronous semi-autonomous agents for integrating design and engineering tools and human specialists in an open environment (as shown in Figure 1). Each tool (or interface for human specialist) can be encapsulated as an agent. These tools and

human specialists are connected by a local network and communicated via this network. Each can also communicate directly with other agents located in the other local networks by the Internet. The agents exchange design data and knowledge via a local network or the Internet via the management agent.

All agents in this system form a group. There are three classes of agents: management agent, tool agents and design agents. These agents are situated on the different layers. The hierarchical relation limits the authority of the agents in the group.

- Management agent locates on the server and manages the whole design group. The action of management agent usually shows the decision and inquiry for the problem, control and supervision for lower layer agents. The knowledge in the KB of a management agent includes all design agent's name, address, and skills or competencies, the history records of performing task and the reward in the group.

- Tool agents include design tools, and management tools. They help management agent to complete system management tasks, such as communication management, task decomposition, database management, knowledge management, collaboration management and system maintenance.

Task decomposition agent help design engineer to decompose a large design task into several sub-tasks.

Collaborative agent matches the sub- tasks and suitable design agents. It also deals with conflict coordination during collaborative design process.

Design tool agents include AutoCAD, Pro-Engineer, Inventor, MicroStation, SolidWorks and so on. It also includes Video Conferencing system for synchronous collaborative design providing run-time support.

Communication agent provides support for interaction among agents and designers by E-mail, text, file, image, graph, audio and video. The exchange of data and files is based on the file transfer protocol (FTP) and TCP/IP protocol.

Process monitor agent watches the whole design process via its event monitor and dynamically maintains the information about the state of each design agent and the status of current design sub-tasks. Whenever a design event happened, the event monitor will be triggered and the correlative message will be passed suitable agents.

Assemble agent checks assembly constraints for finished design components. When constraint violation is found, it will ask collaborative agent and communication agent to solve problem by coordination among design agents.

Knowledge maintenance agent and database agent maintain knowledge base and database respectively.

- Design agents are a kind of domain-dependency agent. They have special design knowledge and ability and can help designers in a special domain.

The creation of complex design in this system is due to collaboration among different agents. These agents contain knowledge of how to design based on their individual strategies and preferences. They are constructed to understand the representation of a design state, be it complete or incomplete, and contribute in a manner that leads to successful solutions. The strategies used by these agents are based on deterministic algorithms. In the current implementation, agents are not autonomous, but are triggered by the system or by other agents.

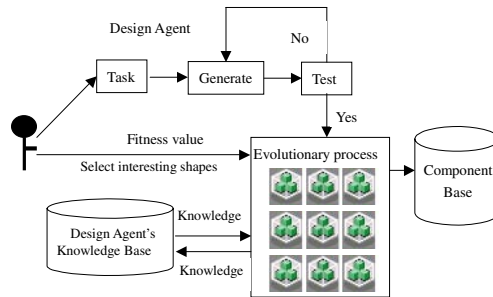


## 4 Design Agent

The majority of agents in the design environment are design agents. A design agent is computer software that in some way helps users to complete design tasks. It is a designer's assistant and can adapt its own ability via interaction with designers and the other agents.

The design agent presented here uses a tree-based genetic algorithm to generate simple 2D sketch shapes, and selected shapes are dealt with to form 3D components. These components are classified and saved in a SQL sever based database (components base).

The design agent uses its knowledge to guide designers during the evolutionary process. Determining new group depends upon designers' interest and the knowledge of the agent (as shown in Figure 2).



**Fig. 2.** A design agent

Computational models of evolution are the foundation of the field of genetic algorithms. Genetic algorithms, originally developed by Holland [23], model the natural selection and process of evolution. Conceptually, genetic algorithms use the mechanisms of inheritance, genetic crossover and natural selection in evolving individuals that, over time, adapt to their environment.

Solving a given problem with genetic algorithm starts with specifying a representation of the candidate solutions. Such candidate solutions are seen as phenotypes that can have very complex structures. The expression of standard generic algorithm has solved many problems successfully. However, when applying genetic algorithms to highly complex applications, some problems do arise. The most common is fixed length character strings present difficulties for some problems. For example, mathematical expressions may be arbitrary size and take a variety of forms. Thus, it would not be logical to code them as fixed length binary strings. Otherwise the resulting algorithm would be restricted and only be applicable to a specific problem rather than a general case. Thus, tree structure, a method useful for representing mathematical expressions and other flexible problems, is presented in this paper.

For a thorough discussion about trees and their properties, see [24]. Here, we only make the definitions involved in our algorithm and these definitions are consistent with the basic definitions and operations of the general tree.

**Definition 1.** A binary expression tree is a finite set of nodes that either is empty or consists of a root and two disjoint binary trees called the left sub-tree and the right sub-tree.

Each node of the tree is either a terminal node (operand) or a primitive functional node (operator). Operands can be either variables or constants. Operator set includes standard operators, basic mathematic functions, triangle functions, hyperbolic functions and so on.

Here we use the expression of mathematical functions in MATLAB (mathematical tool software used in our system). Genetic operations include crossover, mutation and selection. According to the above definition, the operations are described here.

(1) *Crossover*

The primary reproductive operation is the crossover operation. The purpose of this is to create two new trees that contain ‘genetic information’ about the problem solution inherited from two ‘successful’ parents. A crossover node is randomly selected in each parent tree. The sub-tree below this node in the first parent tree is then swapped with the sub-tree below the crossover node in the other parent, thus creating two new offspring.

(2) *Mutation*

The mutation operation is used to enhance the diversity of trees in the new generation thus opening up new areas of ‘solution space’. It works by selecting a random node in a single parent and removing the sub-tree below it. A randomly generated sub-tree then replaces the removed sub-tree.

(3) *Selection*

For general design, we can get the requirement from designer and transfer it into goal function. Then, the fitness value can be gained from calculating the similar degree between the goal and individual by a formula. However, for creative design, it has no standards to form a goal function. In our system, we use the method of interaction with designer to get fitness values. The range of fitness values is from -1 to 1. After an evolutionary procedure, the fitness values that appointed by designer are recorded in the knowledge base for reuse. Next time, when the same situation appears, the system will access them from the knowledge base.

## 5 Assemble Model of Components

After individual work of design agents, some components have been generated, classified and saved in a SQL sever based components base. Assemble agent checks assembly constraints for finished design components. When constraint violation is found, it will ask collaborative agent and communication agent to solve problem by negotiation among design agents.

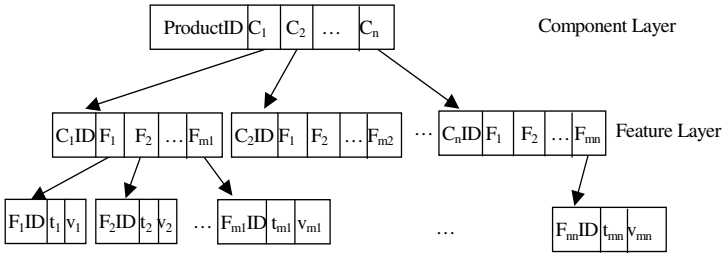
**Definition 2.** Feature  $F_i$  is a tri-tuples  $(F_iID, t_i, v_i)$ , where  $F_iID$  is the name of feature  $F_i$ ,  $t_i$  is the type and  $v_i$  is the value of feature  $F_i$ . In which, value is in broad sense and can be number, character string, array, function, expression, file and so on.

**Definition 3.** Feature vector  $FV$  is defined as a vector  $FV = \langle F_1, F_2, \dots, F_n \rangle$ , where  $F_i$  is a feature.

**Definition 4.** Feature tree FT is defined as  $FT = (D, R)$ , where  $D = \{FV_i\} \cup \text{domain}(FV_i) \cup \{NIL\}$ ,  $FV_i$  is a feature vector and is a node on the feature tree,  $R = \{fri\}$  is a set of relations and constraints among the nodes of the feature tree.

**Definition 5.** Product tree PT is defined as  $PT = (PD, PR)$ , where  $PD = \{FT_i\} \cup \text{domain}(FT_i) \cup \{NIL\}$ ,  $FT_i$  is a feature tree and is a node on the product tree,  $PR = \{pri\}$  is a set of relations and constraints among the nodes of the product tree.

From the above definition, we can discover that the expression of a product can be divided into two layers (as shown in Figure 3) and a multi-branch tree is formed.



**Fig. 3.** The hierarchical structure of a product tree

Genetic operations include mutation, crossover and selection. For simplification, we only describe the operations on feature tree.

(1) *Crossover*

Crossover is implemented by exchanging the sub-tree between two feature trees.

(2) *Mutation*

Due to different encoding strategy, unlike that of traditional genetic algorithms, the mutation operation here is used to make some inventions by changing the nodes and the structure of a feature tree in the following ways: (a) Changing feature value; (b) Changing feature vector such as deleting a feature, adding a new feature etc; (c) Replacing a sub-tree;

(3) *Selection*

In general, more than one required specification exists and all should be taken into consideration when evaluating the solution. If there are  $N$  required specifications  $s_i$  ( $i=1, 2, \dots, N$ ) and  $g_i$  is the proposed solution's specifications, then the distance  $d$  between the required and designed specifications is shown as equation 1.  $\alpha_i$  is the weight value for showing the importance of  $s_i$ . The smallest value of  $d$  would be associated with the best solution for the specified problem.

$$d = \sqrt{\sum_{i=1}^N \alpha_i (s_i - g_i)^2} \quad \text{Equation 1}$$

For creative design, we use the method of interaction with designers to get fitness value. After an evolutionary procedure, the fitness values obtained from the designers are put into the knowledge base of assemble agent for reuse.

## 6 An Architectural Design Example

In this section, we introduce an architectural design example for showing the generative process in our multi-agent design system.

Step 1: Initialize the population of chromosomes. The populations are generated by randomly selecting nodes in the set of operands and the set of operators to form a mathematical expression. We use the stack to check whether such a mathematical expression has properly balanced parentheses. Then, using parsing algorithm, the mathematical expression is read as a string of characters and the binary mathematical expression tree is constructed according to the rules of operator precedence.

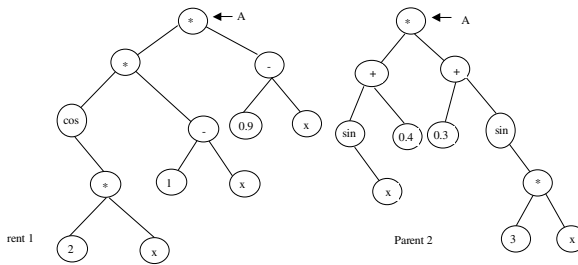
Step 2: Get the fitness for each individual in the population via interaction with designer. The populations with high fitness will be shown in 3D form.

Step 3: Form a new population according to each individual's fitness.

Step 4: Perform crossover, mutation operations on the population.

### (1) Crossover

A crossover node is randomly selected in each parent tree. The sub-tree below this node on the first parent tree is then swapped with the sub-tree below the crossover node on the other parent, thus creating two new offspring. If the new tree can't pass the syntax check or its mathematical expression can't form a normal sketch shape, it will die.



**Fig. 4.** Two parent trees with one crossover node

Taking the two trees in Figure 4 as parent, after the crossover operation on node 'A', we get a pair of children (as shown in Figure 5).

### (2) Mutation

The mutation operation works by selecting a random node in a single parent and removing the sub-tree below it. A randomly generated sub-tree then replaces the removed sub-tree. The offspring will die if it can't pass the syntax check or it can't form a normal shape. Taking the children1 tree in Figure 1 as a parent, one offspring generated by mutation operation is shown as in Figure 6.

After mutation operation, corresponding sketch of parent tree (left) and the generated child (right) are shown as in Figure 7.

Step 5: If the procedure does not been stopped by the designer, go to step 2.

This process of selection and crossover, with infrequent mutation, continues for several generations until the designers stop it. Then the detail design will be done by designers with human wisdom.

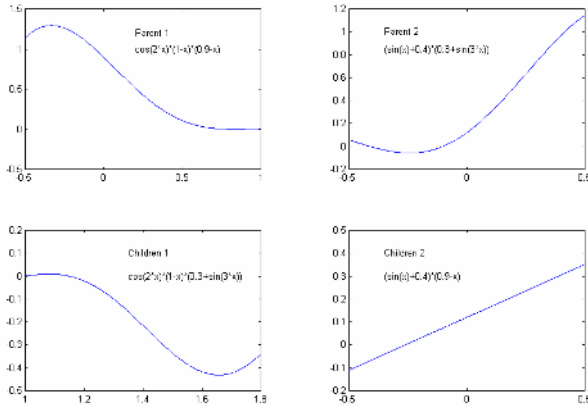


Fig. 5. The results of crossover operation

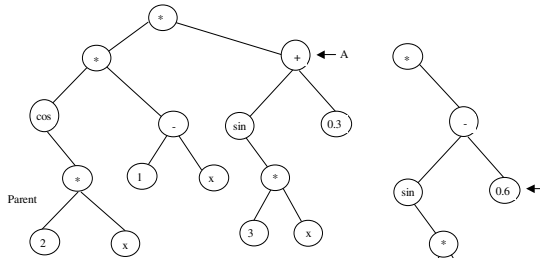


Fig. 6. One parent and a sub-tree for mutation operation

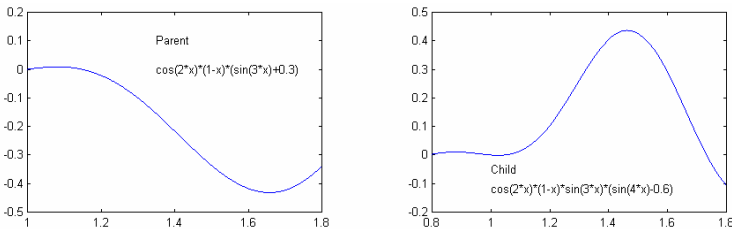
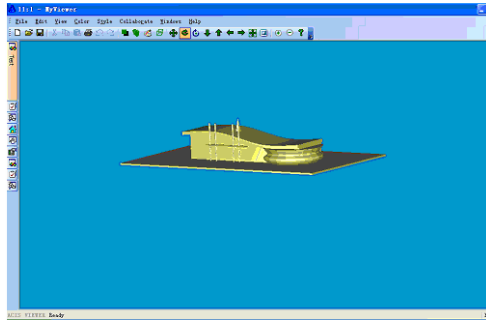


Fig. 7. 2D sketches corresponding to parent and generated offspring trees in Figure 8

Whenever a component design task has been finished, the component will be passed to the assemble agent. The assemble agent checks the components according to their relations and constraints of the design requirements. Conflict will be solved by the design engineer with the help of agents. This process will repeat until all com-

ponents are assembled suitably and the design requirements are satisfied. One assemble result is shown as in Figure 8.



**Fig. 8.** One result generated by the assemble agent through negotiation

## 7 Conclusions

There is still much work to be done before the generative design system can be developed. Our current work is to use the multi-agent architecture as an integrated knowledge-based system to implement a number of learning techniques including genetic algorithms and neural networks. These new algorithms will then be fully integrated with a selected set of 2D (sketching) and 3D (surface and solid modeling) tools and other design support systems. This integrated system is intended for supporting creative design in a visual environment [25].

## Acknowledgements

This project is funded by National Natural Science Foundation of China (No. 69975010, No. 60374054) and supported by Natural Science Foundation of Shandong Province (No. Y2003G01, No. Z2004G02).

## References

1. Schön, D. and Wiggins, G.: Kinds of seeing and their functions in designing. *Design Studies*, 13(2) (1992) 135-156
2. Dewey, J.: The reflex arc concept in psychology, *Psychological Review*, 3, (1896 reprinted in 1981) 357-370.
3. Bartlett, F.C.: *Remembering: A Study in Experimental and Social Psychology*, Cambridge University Press, Cambridge, (1932 reprinted in 1977)
4. Clancey, W.J.: *Situated Cognition: On Human Knowledge and Computer Representations*, Cambridge University Press, Cambridge, England, (1997)
5. Gero, J.S.: Computational models of creative designing based on situated cognition, in T. Hewett and T. Kavanagh (eds.), *Creativity and Cognition 2002*, ACM Press, New York, NY, 3-10, (2002)

6. Clancey, W.J.: *Situated Cognition*, Cambridge University Press, Cambridge, (1997)
7. Gero J.S., *Conceptual designing as a sequence of situated acts*, in I. Smith (ed.), *Artificial Intelligence in Structural Engineering*, Springer-Verlag, Berlin, 1998, pp.165-177.
8. Schön, D. and Wiggins, G.: *Kinds of seeing and their functions in designing*, *Design Studies*, 13(2) (1992) 135-156
9. Suwa, M., Gero, J.S. and Purcell, T.: *Unexpected discoveries and s-inventions of design requirements: A key to creative designs*. in J.S. Gero and M.L. Maher (eds), *Computational Models of Creative Design IV*, Key Centre of Design Computing and Cognition, University of Sydney, Sydney, Australia, (1999) 297-320.
10. Gero, J.S. and Fujii, H.: *A computational framework for concept formation for a situated design agent*, *Knowledge-Based Systems*, 13(6) (2000) 361-368.
11. Kulinski, J. and Gero, J.S.: *Constructive representation in situated analogy in design*. in B. de Vries, J. van Leeuwen and H. Achten (eds), *CAADFutures 2001*, Kluwer, Dordrecht, (2001) 507-520.
12. Smith, G. and Gero, J.S.: *Situated design interpretation using a configuration of actor capabilities*, in J.S. Gero, S. Chase and M.A. Rosenman (eds), *CAADRIA2001*, Key Centre of Design Computing and Cognition, University of Sydney, Sydney, (2001) 15-24
13. Gero, J.S. and Kannengiesser, U.: *Towards a framework for agent-based product modeling*, ICED03, paper no 1621 ICED03 FPC, CDROM (2003)
14. Liu, Y. T. (2000) *Creativity or novelty?* *Design Studies* 21(3): 261-276.
15. Csikszentmihalyi, M.: *Implications of a Systems Perspective for the Study of Creativity*, in R. J. Sternberg (ed.), *Handbook of Creativity*, Cambridge University Press, Cambridge, UK, (1999) 313-335
16. Simon, H. A. (1981) *The Sciences of the Artificial*, MIT Press, Cambridge, MA.
17. Saunders, R. and Gero, J.S.: *Artificial creativity: A synthetic approach to the study of creative behaviour*, in JS Gero and ML Maher (eds), *Computational and Cognitive Models of Creative Design V*, Key Centre of Design Computing and Cognition, University of Sydney, Sydney, (2001) 113-139
18. Frazer, J.H.: *An Evolutionary Architecture*. Architectural Association Publications, London, (1995)
19. Liu, H., Tang, M.X., Frazer, J.H.: *Supporting evolution in a multi-agent cooperative design environment*. *Journal of Advances in Engineering Software*, 33(6) (2002) 319-328.
20. Gero, J.S., Kazakov, V.: *An exploration-based evolutionary model of generative design process*. *Microcomputers in Civil Engineering*, 11 (1996) 209-216.
21. Soddu, C.: *Recreating the city's identity with a morphogenetic urban design*. 17th International Conference on Making Cities Livable, Freiburg-im-Bresgau, Germany, 5-9 (1995)
22. Lander, E. S., *Issues in multiagent design systems*, *IEEE Expert*, 12(2) (1997) 18-26.
23. Holland, J.H.: *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, MI, (1975)
24. Standish, Thomas A.: *Data Structure, Algorithms, and Software Principles*. Addison-Wesley Publishing Company, inc. U.S.A. (1994)
25. Liu, H., Tang, M.X., Frazer, J.H.: *Supporting creative design in a visual evolutionary computing environment*. *Advances in Engineering Software*, 2004, 35(5) (2004) 261-271.

# Similarity Based Agents for Design

Daniel Pinho, Adriana Vivacqua, Sérgio Palma, and Jano M. de Souza

COPPE/UFRJ,

Department of Computer Science, Graduate School of Engineering,

Federal University of Rio de Janeiro,

Rio de Janeiro, RJ, Brazil

dpinho@centroin.com.br, {avivacqua, palma, jano}@cos.ufrj.br

**Abstract.** In this paper, we start from a case study of an architecture company and move on to a framework of agents to identify and inform conceptual design. The processes and problems exhibited by the company involved are quite common and can be found in other similar companies. Our intent is to instrument the current process using agent technology, in an effort to improve global awareness during the conceptual design phase, allowing the designers to design so as to facilitate later steps and optimize the process as a whole.

## 1 Introduction

Technology has been changing at an increased rate. These changes motivate changes at organizational levels. It is now easier to establish communication, exchange information and generally be aware of processes that were hidden. For many companies, however, it has been hard to keep up with the new demands technology makes and to adapt to new work or organizational formats that may improve their performance, without impacting their current business. Companies are struggling to change with as little impact as possible, so as not to compromise their businesses.

In this fashion, even though the organizations may have adopted technology in the daily work environment, it has not been integrated in such a way as to provoke organizational changes and cause true improvement. Most companies still adopt strict organizational models and often times information flows only in one way, causing breaks in communication. For the most part, technology only automates the information flow as it exists.

In a case study of an architecture company, we identified some problem areas that could be addressed and that are present in other segments and companies. The main problem in this type of company is that there are disjoint work groups, and, even though work done by one group (design) defines the work that will be done by the other (physical project), there is little communication between them. There is no feedback from the second group as to what could be improved or what has generated problems for them. This lack of awareness of the project as a whole often generates the waste, delays and problem difficulties.

We have devised an agent-based system to provide a seamless way of integrating the different teams involved and promoting information exchange and awareness of the process as a whole. Agents work with available information about the users' tasks



and their current work and provide information on potential problems of the current design. The intent is to cause as little impact as possible on the way designers work, but to promote changes in their way of designing. Ideally, designers would learn about the consequences of their design choices and about the potential problems they may cause in the later stages of the project, and would design in a more informed way. We will be implementing this system in our case study company and verifying if the new knowledge brought about changes in the designs produced and the designers' way of thinking.

We begin by presenting some background work and then go on to describe our case study, H Camargo Promotional Architecture and Landscaping, examining its processes and information flow. We then go on to describe our approach and the communication agents we are implementing. We wrap up with a brief discussion and conclusions.

## **2 Related Work**

In this section we present some related research that has inspired and guided ours, in particular, agent systems and awareness systems. Computer supported design systems have been the object of much research in the past: ranging from expert and case based reasoning systems to distributed agent approaches, many alternatives have been proposed. A good review of agent based engineering systems can be found in [1].

### **2.1 Agent Systems**

Russel and Norvig define Intelligent Agents as entities that perceive its environment through sensors and act upon it [2]. Agent-oriented techniques are being increasingly used in a range of telecommunication, commercial, and industrial applications, as developers and designers realize its potential [3]. Agents are especially suited to the construction of complex, peer-to-peer systems, because they are lightweight and permit parallelization and easy reconfiguration of the system.

It is currently believed that Multi-Agent Systems (MAS) are a better way to model and support distributed, open-ended systems and environments. A MAS is a loosely-coupled network of problem solvers (agents) that work together to solve a given problem [4]. A comprehensive review of agent systems applied to cooperative work can be found in [5].

### **2.2 Awareness Systems**

Awareness has received a lot of attention among researchers in the past few years, as they start to realize the importance of being aware of collaborators and the environment while working. Initial awareness work focused on video and audio support for cooperation as, for instance in [6] or [7], but other tools and methods have appeared since.

The most basic form of awareness is the one currently provided by messenger systems (such as Yahoo or MSN Messenger, AOL Instant Messenger, etc.). These systems have been widely accepted and adopted.

A more specialized collaborative tool, GROOVE [8] introduces concept of “shared spaces” to increase the scope of personal awareness. In GROOVE’S shared spaces, users can be aware of what others in that space are doing and on what spaces’ objects they are working.

Other researchers have focused on document- or task-based awareness and on providing information to users about who is working on the same document or performing similar tasks at a given moment, as in [9]. Many recent papers address awareness in mobile computing environments, where location awareness is a central issue for collaboration, as in [10] and [11].

More interestingly, some proposals involve motivation, incentives and support for cooperation, such as described in Pinheiro et al. [12]. They propose a framework to provide past event awareness, where users are informed of past occurrences, results and work history of each other (which includes evolution of shared data, members’ actions and decisions, etc.), so as to better collaborate in the present.

Closer to our ideas, Hoffman and Hermann propose a prospect awareness system that allows individuals to envision the potential benefits of collaboration, in an attempt to motivate collaboration [13]. Our system provides potential problem information, in an effort to generate better and more cost-effective designs, avoiding problems in future steps.

### 3 Case Study: An Architecture Company

H. Camargo Promotional Architecture and Landscaping has been a leader in its segment since 1971. It develops custom-made architectural projects for fair and exhibit stands. It is housed in a large pavilion, (with space for administration, workshops and stocks) and has a permanent team of 120 employees. As in any large company, communication problems have started to arise, generating difficulties during project development.

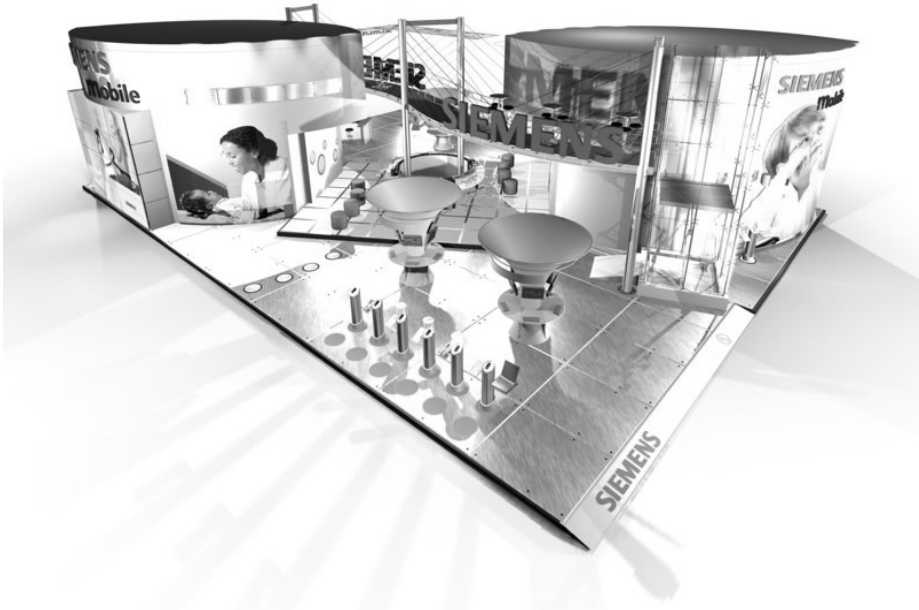
There are four main departments in the company:

- Sales: finds potential clients and their needs.
- Design: creates proposals for these potential clients, establishing the overall designs and some of the materials to be used.
- Project: once a proposal has been accepted, details the project, defining the physical specification: measurements, quantity of materials, how these are to be put together, etc.
- Execution: given the physical specification, executes it, building the actual stand and whatever components may be necessary.

The company essentially functions as two separate entities: the first one (Sales and Design Departments) is responsible for finding new potential clients and designing solutions for those, presenting projects for the stands. The second one (Project and Execution Departments) is responsible for seeing the project through, effectively building the stand to the design initially specified. Stands are all built in-house and

then taken to the event site and put together. Stands already used are either sent to another event or returned to the company for storage.

Project proposals need to be created quickly, be original and innovative. Designs are not charged for, and the company will get paid if the project is accepted (and executed). It is important to note that communication flows almost exclusively in one direction: from the Design Department, a design (a 3D Studio drawing) is handed on to the Project Department and then to Execution. Given that these last two have no say whatsoever in the actual design, oftentimes problems are generated.



**Fig. 1.** Sample design created by designers at H. Camargo

In an effort to create new and interesting designs, architects use materials that currently aren't in stock (and may be hard to purchase) or define shapes that are difficult (if not impossible) to execute, which generates problems for the Project and Execution Teams. Some designs may be harder to implement, which translates to more time spent on the physical specification and difficulties in construction. A sample design is shown in Figure 1.

The construction of a stand, from conception to the moment it is mounted at a fair, involves a series of processes and materials: after approval, a project has to be detailed (further specified) so that it can be mounted in the originally designed way. This specification leads to the use of in-stock materials, and it also creates transformation processes to reuse materials (wood, aluminum and impressions). Some of these transformations are cutting, painting, silking and assembling. In the end, all the pieces have to be arranged in trucks and taken to the fair, where it has to be mounted exactly as initially designed.

Furthermore, this lack of global awareness and communication also increases the possibility of delays in the project (due to difficulties with physical specification and construction), materials waste (if the design includes materials or shapes that cannot be reused), storage of old stands and increase in costs. Naturally, completed projects must be delivered on schedule, which may also lead to a need for overtime or hiring extra personnel to help with construction.

Currently, each team uses computers to perform their part of the process, and hands down files with specifications to the next one. A knowledge base with all the designs created (executed or not) by the company is under construction, and will be used to furnish information to our agents.

In the current model there is a total lack of communication between the teams that design and the teams that build the stand. In many cases this lack of communication and global awareness on the part of the architects generates serious quality problems and makes it hard to reuse of the existing materials in stock.

The great majority of problems are generated when the designer develops a project that demands materials that are not available in stock. In this case, extra costs will be incurred, to purchase materials so that the project is properly executed. In many cases problems occur because the stand is designed without any concern for the way in which it will be constructed. This is an even worse problem, because the project cannot be built in the way it was designed, causing serious quality problems and issues with clients.

Given these issues, we can see that the biggest cause of everything is the lack of awareness and consciousness in relation to other phases of the process. A good designer should be conscientious of all the project phases. Lack of information is a cause of many problems.

## 4 Approach

We have envisioned an agent-based system to inform designers of potential problems during the conceptual design phase. Agents will extract information from each designer's current design and verify the feasibility of this design given previous designs, materials in stock and shapes being utilized.

Our main goal is to provide designers with information on the possible consequences of their current work (for instance, if a certain type of material is out of stock, there is a chance the ordering process will cause a delay in construction). We expect that, given this information, designers will make different decisions, which will benefit the company as a whole.

Information related to the project will be delivered to the designers as the design is developed. Agents have access to stock, process and shape information problems. This information will be used to assess the feasibility and identify possible problems with a project as it is designed.

Agents will analyze the information, link it to other sources of information to establish possible problem points and display this information to the designer as they are designing a new stand. Potential problems are: inexistence of materials, waste, impossibility of reuse and time to construction.

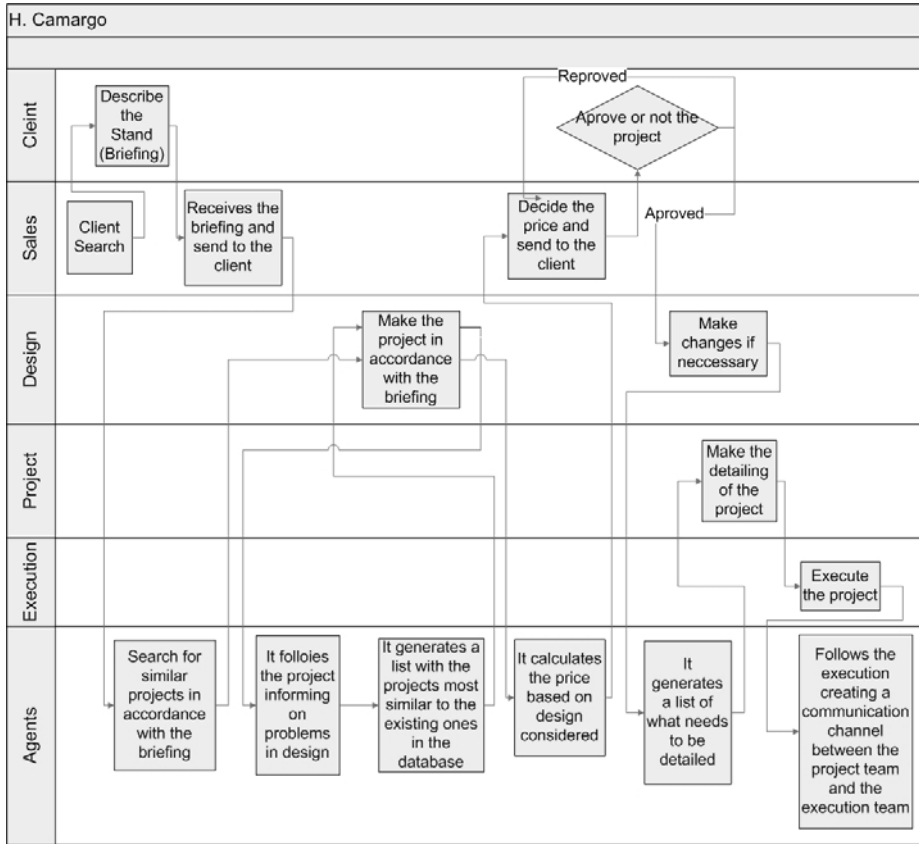


Fig. 2. Envisioned information workflow

- Filtering agents are in charge The agents will have three main functions: filtering, processing and distributing information. of presenting similar projects and extracting the necessary information from the 3d Studio drawing that is being worked on by the designers.
- Processing agents verify whether there are materials in stock or on order that match the information extracted by the filtering agents, and whether some of the old projects or objects can be reused for this one.
- Information distribution agents are in charge of informing the designers about the possible problems with the project and the purchasing Department about the possible need to buy certain materials so that the project can be built as it is designed.

The agent system is being implemented using IBM’s Aglets library. The Aglets library supplies a standard template for agent creation. As mentioned before, there will be three types of agents: filtering, processing and distribution agents.

## 4.1 Filtering Agents

Filtering agents initiate their work on the briefing sent by the client. The information on the briefing usually includes: size of the stand, location, mandatory items, cost and other information about the purpose and about the company. Given this information, a search on the knowledge base provides the previous stands that resemble more closely the briefing at hand. This would create a set of information that would enable the architects to design projects reusing some ideas, shapes and objects and still remain free to experiment with variations on these themes. In the near future we will be collecting this information from the briefing through textual analysis.

Filtering agents also work on 3D Studio files. These files have lots of information about the design. They list the objects used and their location and indicate the vertices for complex shapes. The agent extracts from this file what objects are being used, calculates its size and recognizes its shape from the given vertices.

The following information is extracted from the design file and analyzed by the agents:

- Materials List: a list of all materials that will be used in the construction of the stand. These can be matched against existing materials (in stock) to determine the probability of delays due to lack of material. Note that it is not enough to check with materials currently in stock, but the processing agent has to take into account other designs currently under construction.
- Objects: pre-existing objects (for instance, chairs, desks or stools) that are part of the design. Some of the furniture items may already be in use by other stands. Agents need not only to verify the current snapshot of the information, but to take into account the stands in construction.
- Shapes: shapes used in the construction of each stand or object of the stand. Agents perform shape analysis to see if parts of previous designs can be reused in the current one.

Information collected by the filtering agents will be passed on to the processing agents so it can be analyzed.

## 4.2 Processing Agents

Processing agents evaluate items that exist in stock and shapes under construction. They work with the filtering agents to determine, in real time, if an object or shape can be used in that project, given the expected date of completion. We will be using shape analysis algorithms to assess the viability of the construct. These algorithms are currently under study. These agents are also responsible for determining the costs of materials used and generating a list of materials that will need to be purchased.

The following inferences are made:

- Difficulty in specifying a project or in building certain shapes can be inferred from the time spent on previous similar tasks.
- Possibility for using parts of older stands can be found through shape analysis.
- Furniture reuse can be encouraged by suggesting alternatives, already existing furniture that complies with the overall design (established through shape and color analysis and project history).

The initial list of materials will also help the physical specification teams, as one of their attributions is to generate the complete list of materials, with sizes and quantities. This team also determines which pieces can be reused from other projects and what transformations should be made which is also done by this team.

### 4.3 Distribution Agents

Distribution agents are the interface with the designer. They will pass the information analysis to the designers. It displays messages on the designers' screen, offering useful information during the project.

We are studying what the ideal way of delivering this information is. One possibility is to have a smaller window showing the design and highlighting the problems. Another possibility is to have textual messages describing the problems and icons that display how many problems and what types of problems (structural, materials, time) have been detected. It is important not to draw the designer's attention away from his or her current work too much, for that could compromise their work. We are trying to find an interface that is, at the same time, expressive and unobtrusive.

This agent will also be able to create a direct communication link between the designers and stock team so that a faster analysis can be done. The agent will also offer a communication link between designers and the Execution Department, to clear doubts and create an experience base. These direct links will be implemented in the second phase.

## 5 Conclusions

With filtering, processing and distribution agents we expect to change the way in which designers work: by providing them with data to inform their designs, they will be able to make better design choices, leading to more reuse and fewer errors.

It is important to note that the agents are not meant to restrict the design and do not force the designer into any one solution at any moment. The agent provides awareness information so that the designer can make conscious choices. The designer may still choose to build all-new modules and complicated shapes that cannot be reused, but he or she will be aware of what he or she is doing. We will be investigating the consequences of the introduction of this information at a later time.

The global Knowledge Base under construction will hold information on previous designs (time for construction, objects and materials used, time for assembly, time for physical specification). It will also establish, when possible, a design history and difficulties. Additionally, the agents will store all new information into the knowledge base, creating a case history that can serve as a basis for future inferences.

One addition we would like to make is to provide builders with tools for logging their problems, so that their knowledge can be disseminated throughout the company. Through these, builders would be able to document and create a history of each project. This history would be useful for reflection on the process.

We believe agents are well suited for this type of application and for triggering certain types of organizational changes: they are lightweight and can be easily

integrated with other applications, doing their work without interfering with the designers' work. This type of approach has potential benefits, as it starts to generate a company-wide consciousness that did not exist before, and does so from bottom up, provoking thought, promoting information exchange and increasing process understanding by the designers and architects, instead of having new organizational directives be imposed from top down.

We believe this approach is much more effective than the imposition approach, for individuals can see the consequences of their work, the benefits and problems raised by each design choice. One issue we are especially concerned with is that the application does not limit the designers' creativity, leading to repetitive designs. We will be watching how the introduction of this technology reflects of the designs produced. We hope that designers will still search for novel, creative solutions but that will be more cost effective.

## Acknowledgements

This work was partially supported by CAPES and CNPq grants.

## References

1. Shen, W., Norrie, D.H., Barthès, J.P. Multi-Agent Design Systems for Concurrent Intelligent Design and Manufacturing. Taylor & Francis, London, (2001)
2. Russell, S., Norvig, P. Artificial Intelligence: A Modern Approach. Prentice Hall, Englewood Cliffs, NJ, (1995)
3. Jennings, N.R.: An Agent-Based Approach for Building Complex Software Systems. Communications of the ACM, 44(4) (2001) 35-41
4. Wang, A., Conradi, R., Liu, C.: A Multi-Agent Architecture for Cooperative Software Engineering. Proceedings of the Third International Conference on Autonomous Agents, 1999.
5. Ellis, C.A., Wainer, J.: Groupware and Computer Supported Cooperative Work. Weiss, G. (Ed.) Multiagent, Systems, MIT Press, (1999)
6. Fish, R.S., Kraut R.E., Chalfonte, B.L.: The VideoWindow System in Informal Communications. Proceedings of Computer Supported Cooperative Work (CSCW'90), 1990.
7. Isaacs, E.A., Tang, J.C., Morris, T. Piazza: A desktop Environment Supporting Impromptu and Planned Interactions. Proceedings of Computer Supported Cooperative Work (CSCW'96), Cambridge, MA, (1996)
8. Groove Networks, <http://www.groove.net/home/index.cfm>
9. Morán, A.L., Favela, J., Martínez-Enríquez, A.M., Decouchant, D.: Before Getting There: Potential and Actual Collaboration. Haake, J. M. and Pino, J. A. (Eds.) CRIWG 2002, Springer-Verlag, LNCS 2440 (2002) 147 - 167
10. Aldunate, R. Nussbaum, M., González, R.: An Agent Based Middleware for supporting Spontaneous Collaboration among Co-Located, Mobile and not Necessarily Known People. Workshop on Ad hoc Communications and Collaboration in Ubiquitous Computing Environments, Computer Supported Cooperative Work (CSCW'02) (2002)



11. Esborjörnsson, M., Östergren, M.: Issues of Spontaneous Collaboration and Mobility. Workshop on Supporting Spontaneous Interaction in Ubiquitous Computing Settings, UBIComp'02, Göteborg, Sweden, (2002)
12. Pinheiro, M.K., Lima, J.V. and Borges, M.R.S.: A Framework for Awareness Support in Groupware Systems. Proceedings of the 7<sup>th</sup> International Conference on Computer Supported Cooperative Work in Design (CSCWD'02), Rio de Janeiro, Brazil, (2002) 13-18
13. Hoffman, M, Herrmann, T.: Prospect Awareness - Envisioning the Benefits of Collaborative Work. Available online at: <http://iundg.informatik.uni-dortmund.de/iug-home/people/MH/ProspectAwareness/PAhome.html>

# Semantic Integration in Distributed Multidisciplinary Design Optimization Environments

Ying Daisy Wang<sup>1,2</sup>, Weiming Shen<sup>1,2</sup>, and Hamada Ghenniwa<sup>1</sup>

<sup>1</sup>Dept. of Electrical and Computer Engineering, University of Western Ontario,  
London ON N6G 1H1, Canada

<sup>2</sup>Integrated Manufacturing Technology Institute, National Research Council,  
London ON 6G 4X8, Canada

{ywan6, wshen}@uwo.ca, hghenniwa@eng.uwo.ca

**Abstract.** New design optimization techniques and advanced computer technologies have laid the foundation for the emerging fields of distributed multidisciplinary design optimization (MDO). The challenge now faced by designers is to effectively use this vast amount of knowledge. There are many software tools available in each discipline. However, the key problem is how to integrate these tools and databases, which conform to different interfaces and requirements with little consideration to integration and reusability, in a flexible and robust way. This paper proposes a novel ontology-based semantic integration framework for cooperative distributed MDO environments. The semantic integration aspect will provide the foundation for service-oriented architecture for distributed MDO environments. The cooperation aspect will focus on seamless integration among autonomous MDO systems in dynamic open environments, using multi-agent paradigm.

## 1 Introduction

Multidisciplinary Design Optimization (MDO) is an appropriate methodology for the design of complex engineering systems governed by mutually interacting physical phenomena and made up of distinct interacting software tools. These tools are usually geographically distributed and implemented in different, possibly heterogeneous, computers connected through a network to support complex design projects carried out by multidisciplinary design teams. The recent explosion of the new design optimization techniques and advanced computer technologies has laid the foundation for the emerging fields of MDO. The challenge now faced by designers is to effectively use this vast amount of knowledge. There are many software tools available in each design area. However, the key problem is how to integrate these tools and databases in a flexible and robust way. In particular, most of these tools conform to different interfaces and processing requirements with little consideration to the issues of integration and reusability.

In order to coordinate activities of multidisciplinary design teams and to guarantee the interoperability among the different engineering tools, it is necessary to have efficient collaborative design environments. These environments should not only automate individual tasks, in the manner of traditional computer-aided engineering

tools, but also enable individual members to “share information”, “collaborate” and “coordinate” their activities within the context of a design project. Several features make a multi-agent approach to the integration problem in MDO environments attractive: information is available from many distinct locations; information content is heterogeneous and constantly changing; designers wish to make their work available to the design team, yet retain control over the data; new types of analysis/design and data sources are appearing constantly. To deal with these issues we strongly believe that integration in distributed MDO systems should be supported at the ontology level. This paper reports some of our recent research results based on our previous WebBlow project [20] by applying ontology-based semantic integration. The rest of this paper is organized as follows: Section 2 provides a brief literature review; Section 3 introduces our ontology-based approach for semantic integration in MDO environments; Section 4 presents a case study; Section 5 describes the prototype implementation; Section 6 concludes the paper with some perspectives.

## 2 Literature Review

### 2.1 Semantic Integration

As open, distributed environment spread widely in modern information systems, complexity of system becomes unmanageable. The information is not only distributed, but also heterogeneous. These characteristics require the information to be integrated to provide a unique external appearance and service. The challenge of integration is mainly from the heterogeneity of the information sources. Many types of heterogeneity are due to technological differences [21]. The differences in the semantics of data are one of the causes of these heterogeneities. Semantic heterogeneity occurs when there is a disagreement about the meaning, interpretation, or intended use of the same or related data [21]. Differences in the definition, different precision of the data values and differences in data models are the examples of the cause of semantic heterogeneity. Thus semantic integration plays an essential role among information integration domain.

Researchers and developers have been working on resolving such heterogeneities within the traditional database context for many years. Sheth and Larson [21] defined a reference architecture for distributed database management systems from system and schema viewpoints and discussed the schema level integration. Semantic heterogeneity problem in the context of Global Information Systems (GIS) which are systems geared to handle information requests on the Global Information Infrastructure (GII) can be solved based on the capture and representation of metadata, contexts and ontologies [12]. Hakimpour and Geppert [9] proposed an approach to resolve the issues of interoperability and integration of different data sources caused by semantic heterogeneity based on merging ontologies from different communities. The survey paper Batini et al. [1] discussed and compared 12 methodologies for schema integration. It divides schema integration activities into five steps: pre-integration, comparison, conformation, merging and restructuring. Another survey paper by Rahm and Bernstein [16] discussed and compared 7 prototype schema matchers and 5 related prototypes at schema-level, instance-level, element-level and structure-level, and

language-based and constraint-based criteria. Reck and König-ries [18] presented the similar research work within semi-structure information source domain.

## 2.2 Ontology

Recently, the semantics, which play an important role during the integration task, come into the focus leading to the so-called ontology-based integration approaches. Research on ontology is becoming increasingly widespread in the computer science community. While this term has been rather confined to the philosophical sphere in the past, it is now gaining a specific role in Artificial Intelligence, Computational Linguistics, and Database Theory. A formal and distinct definition of the term “ontology” from the crucial use has been given in [8]. Gangemi et al. [4] further classified ontologies by the degree and type of formalization.

Ontology integration is the construction of an ontology  $C$  that formally specifies the union of the vocabularies of two other ontologies  $A$  and  $B$  [4]. Pisanelli et al. [15] consider ontology integration as the integration of schemas that are arbitrary logical theories, and hence can have multiple models. In order to benefit from the ontology integration framework, we must transform informal schemas into *formal* ones. Wache et al. [22] provided a survey of most prominent ontology-based integration approaches. Pinto [14] further distinguishes three meanings of ontology “integration” as: integration of ontologies when building a new ontology reusing other available ontologies, integration of ontologies by merging different ontologies about the same subject into a single one that “unifies” all of them, and integration of ontologies into applications.

Many of the foundation concepts of ontology have already been established in the areas of intelligent agents and knowledge sharing, such as the Knowledge Interchange Format (KIF), and Ontolingua languages [5]. With the wide acceptance of XML by the Web and Internet communities, XML gained tremendous potential to be the standard syntax for data interchange on the Web. It is also becoming desirable to exchange ontologies using XML. This motivated the development of XML-based ontology languages, such as SHOE [10], Ontology Exchange Language (XOL) [11] and the Resource Description Framework Schema (RDFS) [13]. Other proposals, such as OIL (Ontology Interchange Language) and its successor DAML+OIL [3] attempt to extend RDF and RDFS for ontology.

## 2.3 Agents and Multi-agent Systems

As software agents have grown into a new paradigm for software application development, one question is naturally prompted, indeed, what is an *agent*. There are various definitions of the agent, which describe the agent from different perspectives. One traditional view in AI regards an agent as “anything that can be viewed as perceiving its environment through sensors and action upon that environment through effectors” [19]. A more broad definition of agent is that “an agent is an individual collection of primitive components associated with particular functionalities supporting the agent’s mental state as related its goals” [6].

A general way [23] of defining an agent views the agent as hardware or software-based computer system that enjoys the properties of autonomy, social ability,

reactivity and pro-activity. A stronger notion [23] of agent enforces an agent to have one or more of the characteristics, such as mobility, benevolence, rationality, adaptability and collaboration.

One approach in agent-oriented systems design views the system as a rational agent having certain *mental attitudes* of Belief, Desire and Intention (BDI), representing, respectively, the information, motivational, and deliberative states of the agent [17][2]. The coordinated intelligent rational agent (CIR-Agent) architecture [6][7] is a design paradigm for agents in cooperative distributed systems. The structure of CIR-Agent is based on the mental state as to achieving a goal. CIR-Agent architecture can be employed into many applications, especially in the cooperative distributed systems.

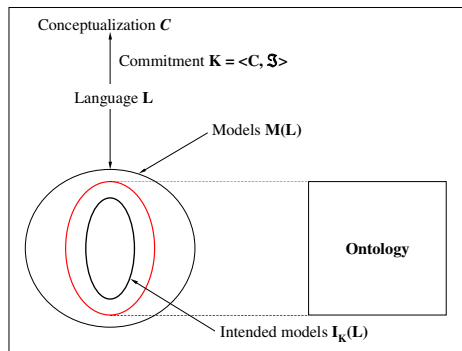
### 3 Semantic Integration for MDO

Both agent technology and Web technology are very useful in developing cooperative design systems. The attractiveness of the Web for propagating information makes it appropriate to integrate with agents for accessing and manipulating information automatically. The challenge is how to build the Web-based environment that will enable and support seamless interaction as well as integration between designers, agents and servers using the available emerging technologies.

To provide an efficient multidisciplinary design environment, it will be essential that integration architecture assumes responsibilities for the semantic level information integration and components interaction, promotes cooperative behaviour among them, and permits users to express their needs as high-level directives as opposed to component-oriented tasks. Our research focuses on the idea of interaction and integration from the perspective of rapidly deployable cooperative distributed systems. The main interaction aspects of multidisciplinary design environment are composition, coordination, cooperation, and adaptation. Composition is the construction of a design solution from a set of entities that might heterogeneous and belong to different disciplines. Coordination is the ability of these entities to manage their interactions in order to solve the design problem. Cooperation is the ability of the entities to share their capabilities and knowledge as related to the problem. Adaptation is the ability of entities to recompose themselves, to improve cooperation with each other, and to re-task their activities, based upon an evaluation of their performance. These main integration aspects laid on the semantic level which enforces an uniform representation for design contributions from various design tools. To deal with the issues of seamless semantic integration, scalability and fault tolerance, we strongly believe that integration in distributed MDO systems should be supported at the ontology level.

Ontology is a collection of vocabulary along with some specification of the meaning or semantics of the terminology within the vocabulary. Ontologies can vary based on the degree of formality in the specification of meaning. The objective is to provide a shared and common understanding of a domain that can be communicated across people and software tools. Further, in the cooperative MDO systems, these ontologies must be integrated to support reasoning that requires the use of multiple ontologies and support interoperability among tools using different ontologies. What is needed is a framework and system architecture that enables agile, flexible, dynamic composition of resources and permits their interaction in a variety of styles to match present and

changing needs of design projects. This framework should go beyond the traditional views of integration; it should provide a level of abstraction at which a multidisciplinary design environment can be viewed collectively as a coherent universe. Theoretically we take the ontology defined by [8]. The term of ontology is defined as a logical theory accounting for the *intended meaning* of a formal vocabulary, i.e. its *ontological commitment* to a particular *conceptualization* of the world. The intended models of a logical language using such a vocabulary are constrained by its ontological commitment. An ontology indirectly reflects this commitment (and the underlying conceptualization) by approximating these intended models [8]. The relationships between vocabulary, conceptualization, ontological commitment and ontology are illustrated in Fig. 1.

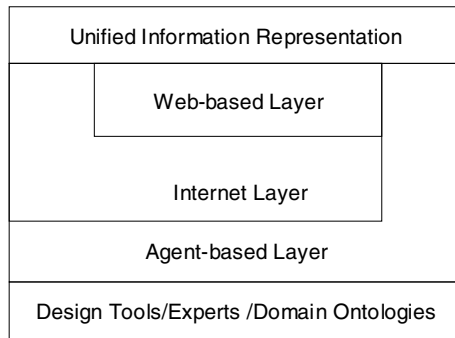


**Fig. 1.** Relationships between vocabulary, conceptualization, ontological commitment and ontology

The primary research effort is to develop an architectural framework for integrating enabling technologies, namely, the ontology-based semantic integration, Internet, Web, and agent-orientation design paradigm for MDO environments. This framework will merge these technologies in a way that each will play the appropriate role in designing an integration environment for MDO. The main characteristics of the proposed architecture are seamless semantic integration, scalability and tolerance for instability in individual systems.

Key issues on the integration in MDO environments include: information is available from many distinct locations; information content is heterogeneous and constantly changing; much of the optimization work for each design aspect is done independently; designers wish to make their findings widely available, yet retain control over the data; new types of design and optimization data sources are appearing constantly. Essentially, however, these problems are familiar from other domains and strongly point to the adoption of agent-orientation approach. Agent-orientation is the next generation modeling and design paradigm for cooperative distributed systems that are autonomous, heterogeneous and capable to interact with each other in a dynamic open environment. In this work we strongly believe that agent-orientation is a very promising design paradigm for such an environment. The first principle of agenthood is

that an agent should be able to operate as a part of a community of cooperative distributed systems environment, including human users. In fact, such a paradigm is essential to model open multidisciplinary design environment, especially considering the multiple dynamic and simultaneous roles a single expert or design tool may need to participate in given design project sessions. In addition, agent technology is rich in the sense of supporting and enabling the automation of complex tasks and developing systems that are reliable and able to take responsibilities of the design project in which they cooperate.



**Fig. 2.** Architecture Framework for Semantic Integration in CDS environment

The proposed agent-oriented architecture framework enables and supports ontology-based cooperative environment. The domain ontology will govern the structural and the behavioural semantics of the design optimization tools in a way that is consistent across all implementations, and is accessible from any implementation. This approach will enable service provisioning from a single technology-independent semantic model (domain ontology) to multiple target component frameworks. Fig. 2 gives an overview of the proposed architecture, showing the relationships between components we are particularly interested in. We provide a global view of design optimization information as a unified data source without knowing about details such as physical location, data structure, etc. The proposed framework allows multiple design optimization information sources and ontologies to be integrated over the Internet through their correspondence or representative agents. It is equally important that the framework supports platform- and language-independent environments as well as “anytime and anywhere” metaphor. To this end, as depicted in Fig. 2, the agents will be enabled through the Internet technology to communicate and coordinate their activities for remote environments. Thus, the design optimization information sources and ontologies will be able to exploit the potentially vast body of knowledge that is distributed over different geographical and computational environments. Whereas the Web-based technology, such as HTML and XML, will complement the Internet by providing the communication harmony between each system components.

## 4 Case Study

### 4.1 Previous Work – WebBlow

The objective of this case study is to build a multi-disciplinary design optimization software environment for the design of automotive interior blow molded parts. The proposed methodology includes distributed system integration using intelligent agents and Internet/Web based technologies; multiple optimization methods including gradient-based optimization techniques and soft computing techniques (neural networks, fuzzy logic, and genetic algorithms, etc.) [20].

### 4.2 Current Work – Ontology-Based Semantic Integration

The extension towards previous WebBlow system is composed of agents, Applets, Servlets and XML databases as shown in Fig. 3. Each of them has own responsibilities and they work together collaboratively. The major entity types in the current design are Interface Agent, BlowDesign Server Agent, Job Agent, EDM Agent, Resource Agent and Semantic Integration Service Agent.

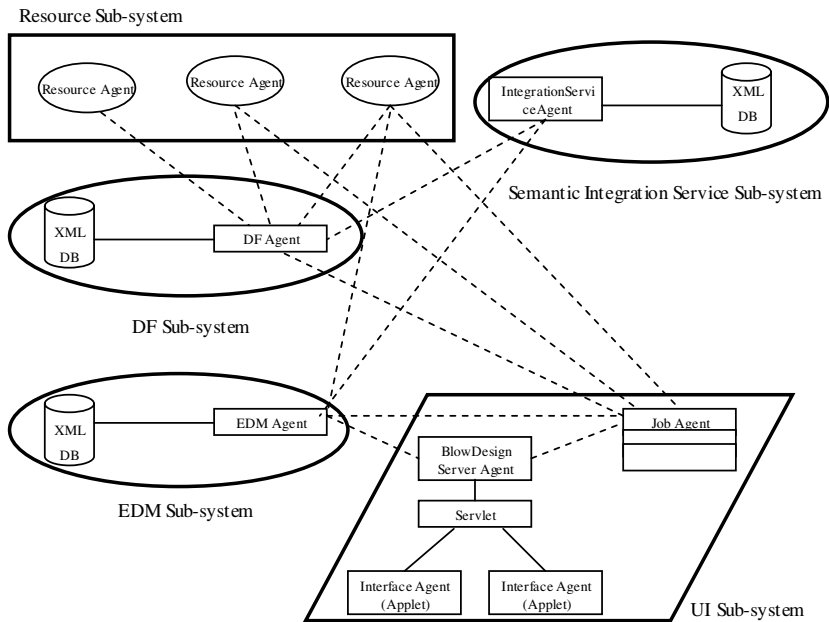


Fig. 3. System Architecture

Interface Agent is responsible of collecting information from the user, displaying results to the user, and validating the data at entry level based on business logic, etc. BlowDesign Server Agent is responsible of receiving requests from user interfaces; communicating with EDM Agent; sending feedback to user interfaces; creating a Job



Agent when the incoming request is an optimization request and all data are valid. Job Agent is created by BlowDesign Server Agent and it communicates with EDM Agent for storing and retrieving data; with DF Agent for finding competent Resource Agents; and with Resource Agents for negotiation based task allocation and job progress monitoring; and finally, it dissolves when the optimization job is accomplished and the results are saved with EDM Agent. EDM Agent is a proactive database agent. Other agents may manipulate or query system administrative data and design optimization project related information about the XML database through EDM Agent. As a middleware agent class, DF Agent (Directory Facilitator) is responsible for agent registration and lookup services within the open environment.

Resource Agent is the major problem solving agent in the MDO environment. Resource agents represent and wrap design optimization tools. They might follow different semantic representation rules for their parameters. However, as long as they claim the ontology that they follow in advance, their optimization calculation would be able to integrate seamlessly by the semantic integration service agents.

Semantic Integration Service Agent is the core part of the system that provides semantic integration services for optimization modules with various information representation standards. First, mechanical ontology is captured and stored in database. The concepts of mechanical domain are modeled. The “part-of” and “is-a” relationship between concepts enables at conceptual level inference. When Resource Agents with different representation rules need to be integrated to accomplish a task, EDM agent would consult with these integration service agents existing in the environment and provide users with a unified representation for the final result. Such kind of service lets user concentrate on domain related issue only rather than pay attention on many irrelevant issues such as data format transformation.

There are three types of databases in the environment. They keep the information for directory facilitator, EDM and ontology respectively. To simplify the problem and focus our effort on semantic aspect of the integration, we use only semi-structured XML file as our data model.

The agents may be physically located within the optimization service provider organization site or anywhere in the world as long as the Internet access is available. Thus HTTP protocol and socket communication are used correspondingly to solve these two types of physical communications.

## 5 Prototype Implementation

The current software prototype is implemented in a network of PCs with Windows NT/2000. The primary programming language is Java, but FORTRAN and C/C++ are also used for legacy software integration. Apache<sup>TM</sup> and Tomcat<sup>TM</sup> are adopted for server side implementation. All Web based user interfaces are implemented using Applets.

At the time of writing this paper, the prototype implementation has been completed. The system consists of three major parts including Web based user interfaces with data collection and transportation, agent-based computing load balancing, and XML based data management and ontological modeling.

## 6 Conclusion

This paper presents an ongoing project on ontology based semantic integration in MDO environments. This research investigates, from a fundamental and practical perspective, several integration and cooperation strategies to develop effective architectures for MDO environments. Also, it attempts to design and develop systems with performance qualities in terms of scalability and robustness.

The prototype has been completed from system requirements definition to system design and implementation. The major work includes ontology-based semantic modeling, semantic integration, Web based user interfaces design and implementation, agent-based computing resource management or load balancing, and XML based data management.

The advantages of the proposed ontology-driven agent-oriented semantic integration architecture includes: (1) the multidisciplinary design optimization problem is cast as a general information gathering problem; (2) multi-agent solution provides mechanisms for dealing with changing data, the appearance of new sources, minding secondary utility characteristics for users, and of course the obvious distributed processing achievements of parallel development, concurrent processing, and the possibility for handling certain security or other organizational concerns.

Future work include: (1) improving the effectiveness of the proposed system for distributed MDO computation in a dynamic open environment; (2) involving more ontologies to further extend the semantic integration up to an ontology integration level.

The semantic integration methodology and general system architecture proposed in this paper can also be used for many other similar application domains such as finance and bioinformatics. Most software modules (e.g., XML based data management, and computing resource management) can be reused in even more other applications.

## References

1. Batini, C., Lenzerini, M., and Navathe, S.: A comparative analysis of methodologies for database schema integration, *ACM Computing Survey*, 18(4) (1986) 323-364
2. Bratman, M.E.: *Intentions, Plans, and Practical Reason*. Harvard University Press, Cambridge, MA, (1987)
3. Fensel, D., van Harmelen, F., and Horrocks, I.: OIL & DAML+OIL: Ontology languages for the Semantic Web. in J. Davis, D. Fensel, F. van Harmelen, (Eds.), *Towards the Semantic Web: Ontology-Driven Knowledge Management*, Wiley, (2002)
4. Gangemi, A., Pisanelli, D.M., and Steve, G.: An Overview of the ONIONS Project: Applying Ontologies to the Integration of Medical Terminologies, *Data Knowledge Engineering*, 31(2) (1999) 183-220
5. Genesereth, M., Fikes, R. (Eds.): *Knowledge Interchange Format, Version 3.0 Reference Manual*, Computer Science Department, Stanford University, Technical Report Logic-92-1, (1992)
6. Ghenniwa, H.: *Coordination in Cooperative Distributed Systems*. PhD Thesis, University of Waterloo, (1996)
7. Ghenniwa, H. and Kamel, M.: Interaction Devices for Coordinating Cooperative Distributed Systems, *Automation and Soft Computing*, 6(2) (2000) 173-184

8. Guarino, N.: Formal Ontology and Information Systems, Proceedings of FOIS'98, Trento, Amsterdam, IOS Press, (1998) 3-15
9. Hakimpour, F. and Geppert, A.: Resolving Semantic Heterogeneity in Schema Integration: an Ontology Based Approach, ACM FOIS'01, Ogunquit, Maine, USA, (2001)
10. Heflin, J., Hendler, J., and Luke, S.: SHOE: A Knowledge Representation Language for Internet Applications, Department of Computer Science, University of Maryland at College Park, Technical Report CS-TR-4078 (1999)
11. Karp, P., Chaudhri, V., and Thomere, J.: XOL: An XML-Based Ontology Exchange Language (XOL version 0.4) <http://www.ai.sri.com/pkarp/xol/xol.html>, (1999)
12. Kashyap, V. and Sheth A.: Semantic Heterogeneity in Global Information Systems: The Role of Metadata, Context and Ontologies, in M. Papazoglou and G. Schlageter (eds.), Cooperative Information Systems: Current Trends and Directions, 139-178, (1998)
13. Lassila, O. and Swick, R.: Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation, (1999)
14. Pinto, H.S.: Some Issues on Ontology Integration, Proceedings of the IJCAI-99 Workshop: Ontologies and Problem-Solving Methods (KRR5), Stockholm, Sweden, (1999)
15. Pisanelli, D.M., Gangemi, A. and Steve, G.: Ontologies and Information Systems: the Marriage of the Century? Proceedings of Lyee Workshop, Paris, (2002)
16. Rahm, E. and Bernstein, P.A.: A survey of approaches to automatic schema matching, VLDB Journal, 10 (2001) 334-350
17. Rao, A.S. and George, M.P.: BDI agents: From theory to practice. Proceedings of the First International Conference on Multi-Agent Systems, Menlo Park, California, (1995) 312-319
18. Reck, C. and Konig-ries, B.: An Architecture for Transparent Access Semantically Heterogeneous Information Sources, 1<sup>st</sup> Int. Workshop on Cooperative Information Agents, (1997) 260-271
19. Russell, S. and Norvig, P.: Artificial Intelligence: A Modern Approach, Prentice Hall, (1995)
20. Shen, W., Wang, Y., Li, Y., Ma, Y., and Ghenniwa, H.: WebBlow: A Web/Agent Based MDO Environment, Proceedings of the Seventh International Conference on Computer Supported Cooperative Work in Design, Rio de Janeiro, Brazil, (2002) 245-251
21. Sheth, A.P. and Larson, J.A.: Federated Database Systems for Managing Distributed Heterogeneous, and Autonomous Databases, ACM Computing Surveys, 22(3) (1990) 183-236
22. Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., and Hübner, S.: Ontology-Based Integration of Information – A Survey of Existing Approaches, IJCAI-01 Workshop: Ontologies and Information Sharing, (2001) 108-117
23. Wooldridge, M. and Jennings, N.: Intelligent Agents: Theory and Practice, The Knowledge Engineering Review 10(2) (1995) 115-152

# Formal Dialogue and Its Application to Team Formation in Cooperative Design

Yisheng An<sup>1,2</sup> and Renhou Li<sup>1</sup>

<sup>1</sup> Institute of Systems Engineering, Xi'an Jiaotong University,  
710049, Xi'an, China  
{aysm, rhli}@mail.xjtu.edu.cn  
<http://www.sei.xjtu.edu.cn/seien/staff.files/lirh.htm>

<sup>2</sup> School of Information Engineering, Chang'an University,  
710064, Xi'an, China  
aysm@chd.edu.cn

**Abstract.** This paper presents a model of formal dialogue between intelligent and autonomous agents and its application to team formation in cooperative design. Using formal dialogue, agents exchange information about their specialty domain, ability, opportunity and willingness. Agents understand each other to form a potential cooperative team, in which joint beliefs and intentions are created. The cooperative tasks can be carried out within the collaborative team, and avoid the blindness in the task decomposition and allocation as usually appeared in cooperative design systems.

## 1 Introduction

Cooperative Design [1] is an application of Computer Supported Cooperative Work (CSCW) [2] in the design and manufacture area. It provides a novel design environment and cooperative mode for designers from different disciplines by using intelligent human-computer interfaces and distributed systems.

In cooperative design, designers, technologists, and experts who are separated geographically and from various disciplines cooperate to perform the products design within a group or a team, in which the traditional centralized approach and system architecture are difficult to meet the requirements of information exchanging, knowledge processing, and data format conversion that are closely related to cooperative process.

In the last decade, a lot of researchers proposed to use agent-based approaches to support cooperative design, such as MetaMorph [3], CoConut [4], CollIDE [5], CADOM [6], and WPDSS [7]. In these approaches, an agent is viewed as an entity that has the perception and interaction ability and is able to solve the problems regarding communication, interaction and knowledge processing in the cooperative design.

The issues associated with agent-based cooperative design include the creation of product and design process model; design knowledge representation; agent and multi-agent system architecture construction; and inter-agent communication.

Another important problem in cooperative design is how to form a team through communicating among agents. However, the team formation is related to how to solve

and understand these issues. Many researchers working in CSCW have paid more attention to solve this problem; mainly, they have been concentrating on two major approaches: joint intention [8] and contract net [9]. Tambe et al. [10] proposed a joint intention based, high reusable model---STEAM, independent of the specific domain knowledge. This model has been used to RobotCup and synthetic transport system in military operation successfully. Rodić et al. [11, 12] proposed a contract net-based INDABA for the implementation of a socially aware embedded multi-agent system, and applied it to RobotCup too.

Compared with above applications, team formation in cooperative design has some distinct properties, which can be described as follows:

- At the initial stage, the potential cooperation partners have not had joint intention and joint belief about the task being performed.
- Since the mankind recognition ability is limited, the reasonable task decomposition and assignment, designer enlistment and resources deployment cannot be conducted before having a comprehensive and clear description of the task.
- The interactive method should be simple, agile and have explicit semantics.

The objective of this paper is to present an overall formal dialogue model and apply it to team formation in cooperative design. Using formal dialogue, agents exchange information about their specialty domain, ability, opportunity and willingness, and understand each other to form a potential cooperative team. Joint beliefs and joint intentions are created for carrying out design tasks.

The remainder of this paper is organized as follows: Section 2 briefly reviews the knowledge base for modeling dialogue and forming cooperative team. Section 3 presents some details of dialogue modeling. Section 4 demonstrates the application of formal dialogue to team formation. Conclusions are presented in Section 5.

## 2 Knowledge Base

### 2.1 Formal Dialogue

Formal dialogue [13, 14, 15, 16, 17] is an interactive and cooperative mode among two or more participants. Each participant explains its changing mental status via talking with each other according to rules defined in advance, and achieves the joint intention for cooperation. Although it roots in philosophy, agent researchers have found that it could be as a novel method for rational interaction between the autonomous agents.

Formal dialogue, as a hopeful inter-agent interaction mode, can be used to eliminate different opinions and conflicts of interests; to work together to resolve dilemmas or find proofs; or simply to inform each of pertinent facts. It overlaps with the work on agent conversation policies [18], but differs in two aspects. Firstly, conversation is a cooperation mode, which postulates the existing inherent joint intention for cooperation, however formal dialogue aims at forming joint intention for cooperation, since each participant only has its own individual intention. Secondly, conversation can be seen as completely free sequences of messages and formal dialogue can be viewed as

a restricted sequence of messages, which satisfies the dialogue policies in specified context.

According to the initially possessed information and the purpose of dialogues, Walton and Krabbe classified them into five categories: Information-seeking Dialogue, Inquiry Dialogue, Persuasion Dialogue, Negotiation Dialogue and Deliberation Dialogue [13, 15]. Formal dialogue, as a further study of the argumentation protocol, is still a research topic in the academic community [14, 15, 16]. Most of the work focuses on the modeling of multi-agent cooperation in electronic commerce [15].

Austin’s speech act theory (SAT) [19] provides the linguistic carrier for the implementation of formal dialogue and other agent communication methods, including KQML and ACL [20]. KQML specifies the format of message and each agent can express desired action and shared knowledge with it. ACL provides different locutions and in addition, gives accurate semantic of each locution. From SAT’s point of view, formal dialogue can be seen as the logic of argumentation between performatives.

### 2.2 Cooperative Design Team

Cooperative design is an embodiment of the behaviors and actions of designers who focus on the task achieving. Since these designers are not only individuals but also belong to a team. Activity of design usually incarnates the associated organization traits. Fulfilling a design task often depends on a set of related activities, namely design targets, and there is a close relationship between activities and resources. Targets related activities often act as a logical unit in design process. Therefore, design process can be viewed as an ordered spatio-temporal distributed logical activity for the specific task, which is undertaken by team members. Cooperative design team may heavily impact on the design efficiency and resource deployment. Two kinds of general team formation process are shown in Fig. 1.

In static cooperation team, agent’s number, position and ability are pre-established and fixed during an entire cooperation process, and members maintain a stable cooperation relationship. Some workflow systems with fixed business process belong to this category.

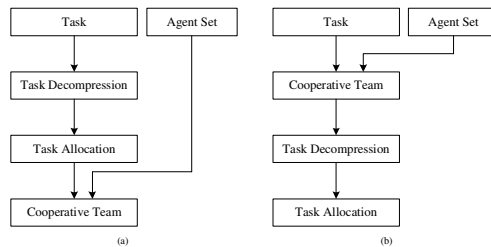


Fig. 1. Two kinds of team forming mechanism

In a dynamic cooperation team, inter-agent relationship will change along with the cooperation process or the transition of internal states. For example, in cooperative design, the relationship between the agent and the initiator is not a simple passive

affiliation, but is an active relation for participating. In addition, the number of agents is not fixed. Each agent's ability, knowledge level, and cognition domain are different. The scale and intensity of cooperation depends on the task and the role and responsibility of each agent.

Before establishing a cooperative team, each agent has its own belief, desire and intention on whether joining in the cooperation or not. Here, joint belief and joint intention will be achieved through uttering—dialogue with specific rules between agents. We give the definition of modal operators used in team formation as follows:

**Definition 1:**  $EBEL_G(\phi)$  represents that every agent in team  $G$  believes  $\phi$ . Its semantic definition is described as:

$$w \models EBEL_G(\phi) \Leftrightarrow \text{iff } \forall i \in G, w \models BEL(i, \phi) \quad (1)$$

**Definition 2:**  $JBEL_G(\phi)$  is true if everyone in  $G$  believes  $\phi$  and everyone in  $G$  believes that everyone in  $G$  believes  $\phi$ .

$$w \models JBEL_G(\phi) \text{ iff } w \models EBEL_G^k(\phi), k \geq 1 \quad (2)$$

**Definition 3:**  $EINT_G(\phi)$  represents that everyone in cooperative team intends to achieve  $\phi$ . Like  $EBEL_G(\phi)$ , its semantic definition is described as:

$$w \models EINT_G(\phi) \Leftrightarrow \text{iff } \forall i \in G, w \models INT(i, \phi) \quad (3)$$

**Definition 4:**  $MINT_G(\phi)$  represents that everyone intends for  $\phi$ , i.e., in order to spurn the competition in team, the agent not only has individual intention but also has mutual intention. Its semantic definition is described as:

$$w \models MINT_G(\phi) \text{ iff } w \models EINT_G^k(\phi), k \geq 1 \quad (4)$$

**Definition 5:** Joint intention  $JINT_G(\phi)$  means that every agent in team mutually intends for  $\phi$  and has joint belief about this mutual intention; it can be formularized as:

$$w \models JINT_G(\phi) \text{ iff } w \models EINT_G^k(\phi) \text{ and } w \models EBEL_G^k(EINT_G^k(\phi)), k \geq 1 \quad (5)$$

### 3 Dialogue Modeling

Formal dialogue is based on the assumption that illocutions executed by agent, like other actions, are for propelling its desire. Speaker must designedly select and organize illocution according to its own desire and assure that illocution would realize its intention effectively. Listener, as another side of the cooperation, must understand speaker's belief, desire and intention through analyzing the speaker's illocution. A dialogue consists of a course of successive utterances, which named *move*.

**Definition 6:** A *move* is a 5-tuple,  $M=(x,y,\delta,\Delta,t)$ , where:

$x$  and  $y$  are speaker and listener of the performative, respectively ;

$\delta$  is an illocutionary operator belonging to the set of {request, statement, assert, concede, challenge};

$\Delta$  is a subject of the performative, which refers either to real-world objects or to states of affairs;

$t$  is a time when performative is uttered. Times are actually timestamps of the related utterance and are modeled to keep track of the evolution of a dialogue.

Based on the passed subject and the illocution transition policies, both parties would select or change illocutionary operators for moving continuously, along with the dialogue evolution. Fig. 2 shows a simple transition diagram for operators, and each transition is represented as an if-then rule as follows:

IF ( $move(x, y, \phi, \Delta, t) \wedge C$ ) THEN  $move(x, y, \phi', \Delta, t + 1)$ ,  $C$  refers to dialogue context.

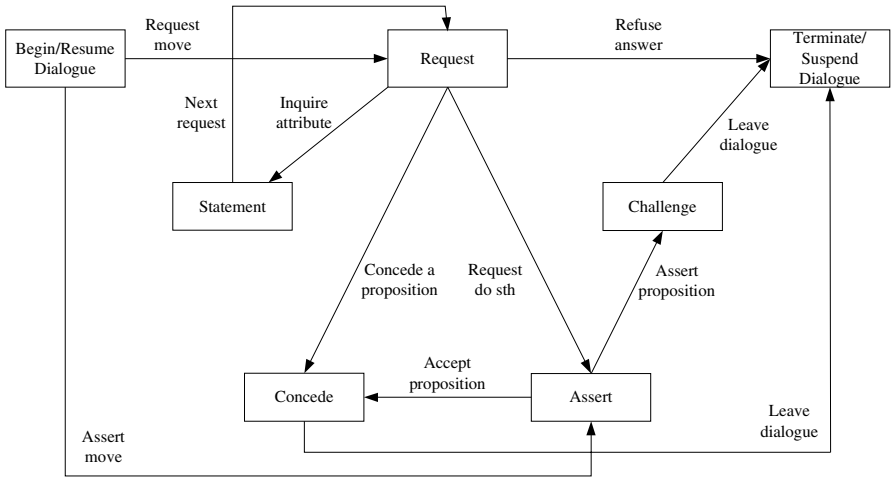


Fig. 2. Illocutionary operators transition diagram for moves

With the above definition, formal dialogue can be viewed as a mathematical form,  $D = (A, \Theta_i, \theta, G, \Sigma, II)$ ,  $i=1, \dots, n-1$ , where  $A$  is a set of agents,  $\Theta_i$  is a context of dialogue from beginning,  $\theta$  is a subject of dialogue,  $G$  the types of dialogue, which belonging to the set of {information-seeking, inquiry, persuasion, negotiation, deliberation},  $II$  is a set of control unit, including open, agree, terminate and nest dialogue,  $\Sigma$  is an order sequence of moves such as:

$$\Theta_0 \mapsto m_1, \Theta_1, m_1 \mapsto m_2, \dots, \Theta_{n-1}, m_{n-1} \mapsto m_n \tag{6}$$

### 4 Application of Formal Dialogue to Team Formation

This section demonstrates the application of formal dialogue to the team formation in cooperative design, including joint intention forming process expressed in modal



logic and a practical dialogue process. Some symbols and proposition are described as the following:

$\varphi$ , a co-design task;

$\phi$ , a proposition which take “achieving co-design task  $\varphi$ ” as main goal;

$a$ , an initially undertaker of co-design task;

$L = \{ag_1, ag_2, \dots, ag_n\}$ , a set of agents;

$Lc^\varphi = \{ag_1^\varphi, ag_2^\varphi, \dots, ag_m^\varphi\}$ , a multi-agent cooperation team that undertakes  $\varphi$ .

For the co-design task  $\varphi$ , the initiator  $a$  regards proposition  $\phi$  (executing the co-design task  $\varphi$ ) as its intention, but the initiator  $a$  is unable to accomplish the task by itself; it hence searches for potential cooperators  $ag_i$  ( $ag_i \in L$ ), and forms a cooperative team to accomplish the task together.

During the cooperative design process, it is insufficient that every agent only has its individual intention. Except for calculating its own action planning, agent also pays attention to others' behavior, and under the prerequisite that it is helpful for complementing the whole design task; it should be able to change its own action planning according to other agents' request. After having clearly known the potential cooperative agent's specialty domain, ability, willingness and intention for joining in the design task, the forming process of the co-design team can be viewed as a joint intention forming process. The detailed process is explained as follows:

At the first stage, the initiator opens an information-seeking dialogue with each agent  $ag_i$  ( $ag_i \in L, 1 \leq i \leq n$ ) in turn, inquires the specialty domain, ability, opportunity and willingness for joining in a task  $\varphi$ , and judges whether an agent is suitable. The process can be described as follows:

M1: ( $a, ag_i, request, "specialty domain of ag_i", t$ );

M2: ( $ag_i, a, statement, "vehicle engineering", t+1$ );

M3: ( $a, ag_i, request, "wil(ag_i, \varphi)", t+2$ );

M4: ( $ag_i, a, assert, "wil(ag_i, \varphi)", t+3$ ) or  
( $ag_i, a, assert, "\neg wil(ag_i, \varphi)", t+3$ );

Once  $a$  receives M4, it will use a rule to update its beliefs:

If  $trust(a, wil(ag_i, \varphi))$  then  $bel(a, wil(ag_i, \varphi))$ .

According to the similar procedure, the initiator inquires the ability and opportunity of participating task  $\varphi$ , and will use the following rules to update its beliefs either:

If  $trust(a, able(ag_i, \varphi))$  then  $bel(a, able(ag_i, \varphi))$

If  $trust(a, opp(ag_i, \varphi))$  then  $bel(a, opp(ag_i, \varphi))$

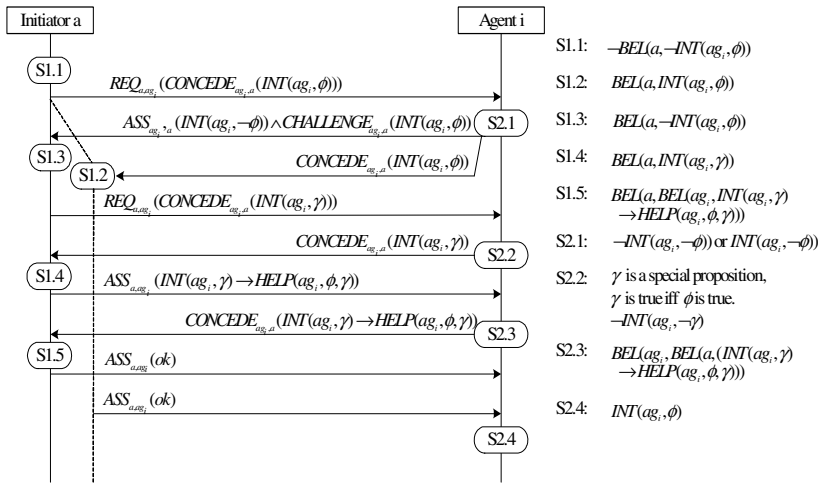
Above inferences process can be given in the following dynamic logic expression:

$$\begin{aligned} & [(a, ag_i, request, "if wil(ag_i, \varphi) then (ag_i, a, assert, "wil(ag_i, \varphi)", t) \\ & else (ag_i, a, assert, "\neg wil(ag_i, \varphi)", t)", t); (ag_i, a, assert, "wil(ag_i, \varphi)", t)] \\ & trust(a, wil(ag_i, \varphi)) \rightarrow bel(a, wil(ag_i, \varphi)) \end{aligned} \quad (7)$$

The dynamic logical expression  $[\alpha; \beta]\phi \leftrightarrow [\alpha][\beta]\phi$  is used to express that the formula is true after sequentially executing  $\alpha$  and  $\beta$ . Here,  $t$  is a placeholder for time. In the same way, the ability and opportunity of  $ag_i$ , who will undertake task  $\phi$  can be sought.

At the second stage, the initiator opens a persuasion dialogue for asking each agent  $ag_i$  ( $ag_i \in L, 1 \leq i \leq n$ ) to take same intention, namely regarding  $\phi$  (realizing design task  $\phi$ ) as its own intention, which is represented as  $INT(ag_i, \phi)$ .

The process that the initiator persuades  $ag_i$  to accept  $\phi$  (achieving design task  $\phi$ ) as its intention is shown in Fig. 3. Each move in this diagram has an alternative format, which picks up the illocutionary operators as 'moves' name.



**Fig. 3.** Initiator a persuade agent i regarding  $\phi$  as intention

At the third stage, the initiator may ask each agent to accept the preposition  $MINT(ag_i, \phi)$  as its own intention, written as  $INT(ag_i, MINT(ag_i, \phi))$ , namely agents mutually intend to achieve design task  $\phi$ . Similar process showed in Fig. 2 can be applied to the formation process of  $INT(ag_i, MINT(ag_i, \phi))$ .

Integrating the two formulas given above,

$$INT(ag_i, \phi) \wedge INT(ag_i, MINT(ag_i, \phi)) \rightarrow INT(ag_i, \phi \wedge MINT(ag_i, \phi)) \quad (8)$$

With the definition 3, the following expression is obtained.

$$INT(ag_i, \phi \wedge MINT(ag_i, \phi)) \rightarrow EINT(\phi \wedge MINT(ag_i, \phi)) \quad (9)$$

After all the agents in the potential cooperative team receive the above intention that is confirmed in joint belief, the resulting joint intention is written as  $JBEL(ag_i, EINT(\phi \wedge MINT(ag_i, \phi)))$ .

Now, taking a practical dialogue process to illustrate the information seeking and persuasion stage in team formation with dialogues occurrence between the initiator (I) and a potential partner (A). Let the initiator receives a task for designing a crankcase, which is unable to be accomplished by it whose goal is to establish a team. To achieve this goal, the initiator contacts with agent i through information-seeking dialogue.

#### Information-Seeking Dialogue:

```
I: BEGIN (INFOSEEK (crankcase design))
A: AGREE (INFOSEEK (crankcase design))
I: request (ability, opportunity, willingness)
A: SUSPEND (INFOSEEK (crankcase design)) AND BEGIN (INFOSEEK
  (task detail))
I: AGREE (INFOSEEK (task detail))
  A: request (specialty domain)
  I: statement ("vehicle engineering")
  A: request (degree)
  I: statement ("difficult")
  A: request (timeliness)
  I: statement ("as soon as possible")
A: CLOSE (INFOSEEK (task detail))
A: statement (85,80,70)
I: CLOSE (INFOSEEK (crankcase design))
```

Maybe agent i is the most competent candidate, but unwilling to do it for some reasons. Therefore, a persuasion dialogue will be opened.

#### Persuasion Dialogue ( $\phi$ ---participating in the crankcase design):

```
I: BEGIN (INFOSEEK (A takes  $\phi$  as intention))
A: AGREE (INFOSEEK (A takes  $\phi$  as intention))
I: request (attitude toward takes  $\phi$  as intention)
A: statement ("no")
I: SUSPEND (INFOSEEK (A takes  $\phi$  as intention)) AND
  BEGIN (PERSUASION (A should take  $\phi$  as intention))
A: AGREE (PERSUASION (A should take  $\phi$  as intention))
  I: assert (A must take  $\phi$  as intention)
  A: challenge (A must take  $\phi$  as intention)
  I: SUSPEND (PERSUASION (A should take  $\phi$  as inten-
    tion)) AND BEGIN (INFOSEEK (current work))
  A: AGREE (INFOSEEK (current work))
    I: request (work)
    A: statement ("flywheel", "hubcap", "camshaft")
  I: CLOSE (INFOSEEK (current work)) AND
    RESUME (PERSUASION (A should take  $\phi$  as intention))
  I: assert (camshaft design can be included in crank-
    case design)
  A: concede
  I: assert (stop current camshaft design activity and
    add crankcase design)
  A: concede
I: CLOSE (PERSUASION (A should take  $\phi$  as intention))
I: CLOSE (INFOSEEK (A takes  $\phi$  as intention))
```

The example demonstrates most of the abilities of the proposed model. More specifically, we have showed how agents take the activities to start a discussion and act autonomously to conclude it. The following three features should also be noted through above example. Firstly, the dialoguing agent is autonomous and honest, and it will not blindly undertake any actions that conflict with its mental status and misguide

others. Secondly, the agent is willing to participate in and cooperate with other team members for the global task. Under the prerequisites of accomplishing the global task, it can modify its local action plan and mental state for executing other agents' task. Thirdly, at any time and any circumstances, agent can terminate or suspend a dialogue without any permission by others.

## 5 Conclusion

The main contribution of this paper is to extend an inter-agent communication method, which was first proposed by Walton and Krabbe in [13]. After introducing the knowledge base of formal dialogue, a formal dialogue model has been proposed and applied to the team formation in a cooperative design environment. Under the proposed model, the intention of achieving an individual task can be converted into the joint intention and joint action of the designers. The process makes full consideration of the agent's specialty domain, ability, opportunity and willingness.

The merit of using the proposed dialogue model in team formation in cooperation design can be described as follows:

- 1) Contract Net [9] maps the team forming process and task assignment to bid. Despite its simplicity and effectiveness, it makes a rigorous restriction that each agent has joint intention in a group and initiator must have a comprehensive and clear description of the task. Therefore, the proposed dialogue model is more flexible than The Contract Net for team building.
- 2) The proposed approach is also better than other agent conversation policies approaches because it is easier to describe how to speak and what to say. The essential difference between them is that the proposed dialogue model not only expresses the attitude of the given illocution, but also contains the logical reasoning process with attitudes.

## References

1. Schmidt, K.: Cooperative design: Prospects for CSCW in design. *Design Science and Technology*, 6(2) (1998) 5-18
2. Grudin, J.: Computer-Supported Cooperative Work: History and Focus. *IEEE Computer*, 27(5) (1994) 19-26
3. Shen, W., Maturana, F., Norrie, D.H.: MetaMorph II: an agent-based architecture for distributed intelligent design and manufacturing. *Journal of Intelligent Manufacturing*, 11(3) (2000) 237-251
4. Stork, A., Jasnoch, U.: A Collaborative Engineering Environment. *Proceedings of Team-CAD Workshop on Collaborative Design*, (1997) 25-33
5. Nam, T.J., Wright, D.K.: CollIDE: A Shared 3D Workspace for CAD. *Proceedings of the 4th International Conference on Networking Entities*, (1998) 103-105
6. Rosenman, M., Wang, F.J.: CADOM: A Component Agent2based Design2Oriented Model for Collaborative Design. *Research in Engineering Design*, 11(4) (1999) 193-205
7. Qiang, L., Zhang, Y.F., Nee, A.Y.C.: A Distributive and Collaborative Concurrent Product Design System through the WWW/ Internet. *International Journal of Advanced Manufacturing Technology*, 17(5) (2001) 315-322

8. Cohen, P.R., Levesque, H.J.: Teamwork. Special Issue on Cognitive Science and Artificial Intelligence, 25(4) (1991) 487-512
9. Smith, R.G.: The Contract-net protocol: high-level communication and control in a distributed problem solver. *IEEE Transactions on computers*, 19(12) (1980) 1104-1113
10. Tambe, M.: Towards flexible teamwork, *Journal of Artificial Intelligence Research*, 7 (1997) 83-124
11. Rodic, D., Engelbrecht, A.P.: INDABA-Proposal for an Intelligent Distributed Agent Based Architecture. *Proceedings of the Second International Conference on Computational Intelligence, Robotics and Autonomous Systems*, (2003)
12. Rodic, D., Engelbrecht, A.P.: Investigation into the Applicability of Social Networks as a Task Allocation Tool for Multi-Robot Teams. *Proceedings of the Second International Conference on Computational Intelligence, Robotics and Autonomous Systems*, (2003)
13. Walton, D.N., Krabbe, E.C.W.: *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*, SUNY Press. (1995)
14. McBurney, P., Parsons, S.: Dialogue Games in Multi-Agent Systems. *Informal Logic, Special Issue on Applications of Argumentation in Computer Science*, 22(3) (2002) 257-274
15. McBurney, P., Parsons, S.: Games that agents play: A formal framework for dialogues between autonomous agents. *Journal of Logic, Language and Information*, 11(3) (2002) 315-334
16. McBurney, P.: *Rational Interaction*. Ph.D Thesis, University of Liverpool, (2002)
17. Parsons, S., Wooldridge, M.: Properties and complexity of formal inter-agent dialogues. *Journal of Logic and Computation*, 13(3) (2003) 347-376
18. Chaib-Draa, B., Dignum, F.: Trends in agent communication language. *Computational Intelligence*, 18(2) (2002) 89-101
19. Traum, D.R.: Speech Acts for Dialogue. In Wooldridge M., Rao A. (eds.), *Agents Foundations of Rational Agency*, Kluwer, (1999) 169-201
20. Wooldridge, M.: *An Introduction to Multi Agent Systems*, John Wiley & Sons, 2002

# MA\_CORBA: A Mobile Agent System Architecture Based on CORBA

Xingchen Heng<sup>1,2</sup>, Chaozhen Guo<sup>1,\*</sup>, and Jia Wu<sup>1,3</sup>

<sup>1</sup> Mathematics and Computer Science College, Fuzhou University,  
Fuzhou, Fujian, P.R. China, 350002

<sup>2</sup> Research Institute of Computer Software, Xi'an Jiaotong University,  
Xi'an, Shanxi, P.R. China, 710049

<sup>3</sup> School of Computing Science and Engineering,  
University of Salford,  
Salford, M5 4WT, UK

guocz@263.net, boyhxc@163.com, J.Wu@pgt.salford.ac.uk

**Abstract.** Both CORBA and mobile agent technologies have caught a wide attention in research and development fields. Integrating the two technologies to make CORBA objects movable is the main idea of this paper. A CORBA based mobile agent server architecture, called MA\_CORBA, is proposed and its prototype is presented.

## 1 Introduction

The software architecture, developed in 1990's, is a kind of abstract software technology mechanism that describes the whole organized structure and capability of software systems. With increasing applications of distributed systems technology, people pay more and more attention to the design of distributed middleware architectures. CORBA (Common Object Request Broker Architecture) is a representation of distributed middleware architectures. Mobile agent system is a new distributed computing and network communication model.

Nowadays, the usage of CORBA and mobile agent technologies can be approximately divided into two categories: (1) using CORBA and mobile agents separately, i.e., using them in different mobile agents without considering their interrelation; (2) using CORBA and mobile agents at the same time and considering their interrelation. The latter one is one of the main tasks of mobile agent system researchers with support from powerful technology and marketing priority of CORBA/IIOP (Internet Inter-ORB Protocol). Our work mainly concentrates on the cooperation of mobile agents and CORBA objects. In some extent, CORBA and mobile agent technology can make up the shortage of each other. However, these two technologies are still separated, and have their own running environments, objects and etc. At present, a few reports discuss integrating mobile agents with CORBA at the architecture level. No one has introduced CORBA's lifecycle management and naming server manage-

---

\* Corresponding author.

ment into mobile agent architecture, or makes it as the primary model. This paper brings forward a CORBA based mobile agent server architecture, called MA\_CORBA (Mobile Agent\_CORBA), in order to make objects in CORBA server full of characteristics of mobile agents [1, 2].

The remainder of this paper is organized as follows. Section 2 outlines the character of mobile agents and CORBA and presents MA\_CORBA, a server architecture integrating mobile agent and CORBA technologies. Section 3 describes the prototype implementation and the performance evaluation of MA\_CORBA. Finally, the paper is concluded in Section 4.

## 2 System Architecture of MA\_CORBA

### 2.1 Introduction to MA\_CORBA

CORBA has been applied successfully in many areas such as enterprise management, telecom and so on. The software architecture that follows CORBA standard is important in constructing distributed computing systems. But CORBA still has some inherent limitation. The most typical one is that it has not radically broken away from the limitation of remote invocation and does not support code transfer (migration). At the same time, communication cost is very high.

But the code transfer is the basic characteristic of mobile agents. Agent technology, especially mobile agent technology provides a new method of analyzing, designing and realizing distribute systems. It is considered as “another great breakthrough of software development” [3, 4]. Agent technology has been used in various application domains. With the development of the Internet/WWW technology, the research of mobile agents is becoming more and more popular. Mobil agent technology breaks through the limitation of remote invocation completely and realizes the independent mobility of agents. On the other hand, COBRA is mature in object technology, but it has not realized the real independence on the interface of client application and object realization. The mutual operation and reuse of components still stay at the code level. The application of mobile agents with the ability of running in different platforms and self-control in different structures is a distributed computing objective. The characteristic of encapsulation, inheriting and reuse makes it rise to the semantic and knowledge level.

Therefore, using mobile agents’ independent migration in CORBA can greatly enhance CORBA object abilities of transferring. On one hand, mobile agent technology can make up CORBA’s shortcoming in application integration, mutual operation and reuse. On the other hand, the CORBA standard’s flexible extension and security guarantee and shielding bottom platform, provides an ideal infrastructure for the implementation of mobile agents’ automatic transfer, ways of transfer, security and so on. Combining the two technologies, the proposed CORBA based mobile agent server architecture provides a new development space for CSCW (Computer Supported Cooperative Work) mutual operations, disclosure and extension in heterogeneous environments [5,6].

### 2.2 MA\_CORBA System/Server Architecture

MA\_CORBA system includes user interface, MA\_CORBA Object, MA\_CORBA support environment and Java Virtual Machine. Users can create and manage MA\_CORBA objects through the user interface, which adopts advanced technology of user-computer interface to satisfy each user’s special needs. MA\_CORBA Object is a CORBA object which is created according to the user’s request, applied for special purpose and transferable. It can carry out the special task in the MA\_CORBA support environment. The MA\_CORBA support environment is the core of MA\_CORBA system architecture, including transfer service, directory service, and communication service. These four parts and relations between them make up of MA\_CORBA system architecture as shown in Figure.1.

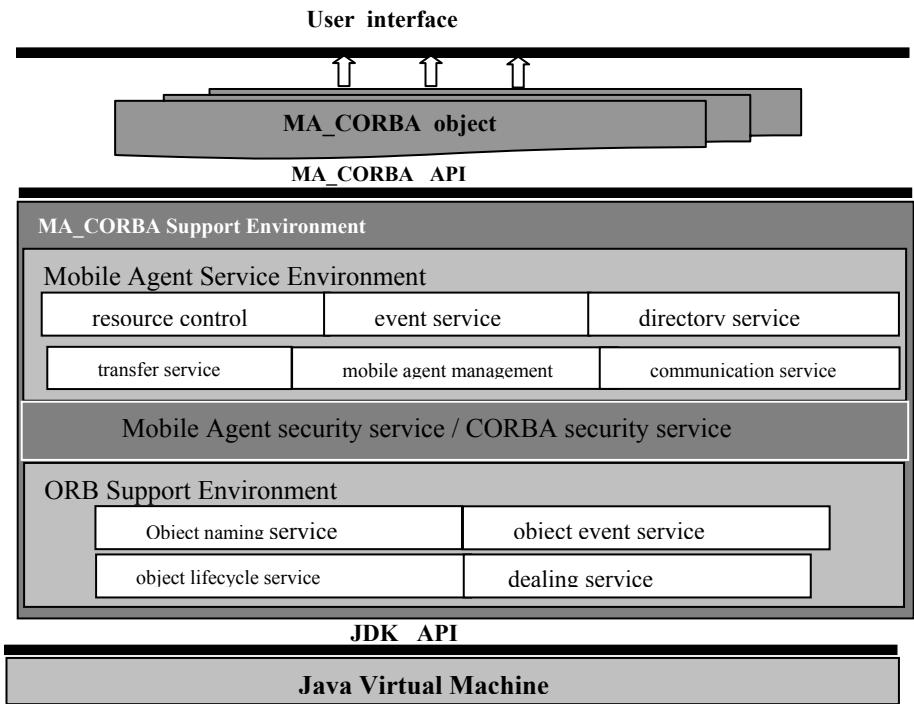


Fig. 1. The MA\_CORBA system/server architecture

MA\_CORBA architecture has three layers and three interfaces. The bottom layer is Java Virtual Machine (JVM), and it is the running environment for MA\_CORBA objects and supporting environment. It provides basic programming interface for programmers with JDK (Java Development Kit) API. The middle layer is MA\_CORBA supporting environment, and it is the core of the MA\_CORBA system architecture. It provides programming interface for the higher layers. The highest layer is user interface. The MA\_CORBA supporting environment includes two parts:



mobile agent service environment and ORB support environment. The main components in the MA\_CORBA supporting environment are described as follows [5.6]:

### **1) Transfer service**

Mobile agent services environment adopts ATP [7,8] (Agent Transfer Protocol) which is provided by IBM Research Center in Japan to implement MA\_CORBA object's transfer between host computers, set up remote running environment and all kinds of service interfaces.

### **2) Communication service**

Mobile agent service environment adopts ACL (Agent Communication Language) [1,9] to implement communication between MA\_CORBA objects, or between MA\_CORBA object and service environment. Because of the communication among MA\_CORBA objects in the same server, every MA\_CORBA object pass messages through cooperation mechanism based on CORBA specification. Every MA\_CORBA object supplies some methods to deal with all kinds of external messages. And communication between them will be implemented by calling each other's involved methods of processing messages.

### **3) Resource control service**

Resource is the floorboard of service and messages that are supplied by server, including all kinds of hardware, software and data. For utilizing these resources on the server, mobile agents move to the server. Therefore, all resources must be managed securely and reliably. This issue is addressed using the concept of proxy which is often used in network service. The proxy has the same interface as resource. MA\_CORBA object does not acquire the resource's references directly, and MA\_CORBA object's service requests are all sent to the proxy. The proxy will judge the requests, and if the request is within the security limitation, the request will be sent to the corresponding resource, otherwise the proxy refuses the request directly.

### **4) Directory service**

Directory service provides lookup services. It makes MA\_CORBA objects in server communicate with remote host computer and MA\_CORBA objects and utilizes all kinds of external services. Directory service is based on the CORBA's naming service and dealing service. Naming service supplies the ability of binding a name with an object and identifying the object from an object context in terms of name. Dealing service resembles yellow pages and supplies the ability of lookup in terms of object types and properties.

### **5) Mobile agent management service**

It is an important component of the mobile agent server environment. It allocates all servers which are needed by MA\_CORBA objects. For example, it allocates the server of MA\_CORBA objects' identification to security control module, allocates task of transferring MA\_CORBA objects to transfer service module, allocates task of communication between MA\_CORBA objects to communication service module and coordinates every module's running properly.

### **6) Security service**

Security is a universal and important problem of mobile agent systems. The security service in the architecture model is based on CORBA's security service, and expands

agent's security service mechanism. Because agent and server cannot prefigure exactly the aftereffect of their behaviors, the uncertainties bring severe security problem. In an agent system, security mechanism is bi-directional. In one aspect, it assures an agent itself not to be destroyed, and in another aspect, it assures server not to be destroyed by the agent through encrypting, data-signing, security authentication, and so on.

### **7) Event service**

The event service supplied by the mobile agent service environment is based on CORBA's mature event service mechanism. Event service supplies flexible, powerful and configurable basic ability, and manages registration and information events which are sent to MA\_CORBA objects or given out from MA\_CORBA objects. So, a wide-coupling communication channel is set up among MA\_CORBA objects which do not know each other. The basic event service includes MA\_CORBA objects' perception and response to the external environment.

### **8) ORB support environment**

It realizes mainly four kinds of basic services of CORBA: object lifecycle service, object naming service, object event service and object dealing service. All these services are running in ORB environment. When an object starts up for the first time, it needs to find other objects in ORB, which is the function of naming service. Lifecycle service maintains object life period, such as life reservation, termination and relocatability, so users can establish, convey, and terminate object. Event service offers the asynchronous interaction between the anonymous objects. Users will get the notice when an important thing happens in ORB (Object Request Broker).

## **3 Prototype Implementation and Performance Evaluation**

### **3.1 MA\_CORBA Prototype Implementation**

MA\_CORBA prototype system is a CORBA platform which is based on mobile agent development/server platform. This system runs in our lab's local network. The implementation tools used for prototype implementation include Borland Jbuilder 7.0, Inprise Visibroker for Java 4.5, and IBM Aglet1.1. Therefore, CORBA supplies necessary service for mobile agent manager and basic facilities for agent-agent, agent-manager, and agent-user communication. Figure. 2 shows a schematic view of the prototype system.

MA\_CORBA system mainly includes two parts: (1) MA\_CORBA service environment including Aglet 1.1 service process and Visibroker 4.5 service process; (2) MA\_CORBA objects which are CORBA objects with the characteristics of transferring freely. MA\_CORBA service environment is in charge of creating, suspending, transferring, resuming, executing and terminating MA\_CORBA objects. In terms of MASIF (The OMG mobile Agent system interoperability facility), a host computer may include one or more agent systems; an agent system may include one or more places (context running environment) [1, 10] where agents run; and a place may include one or more agents. In the implemented MA\_CORBA system, the class "

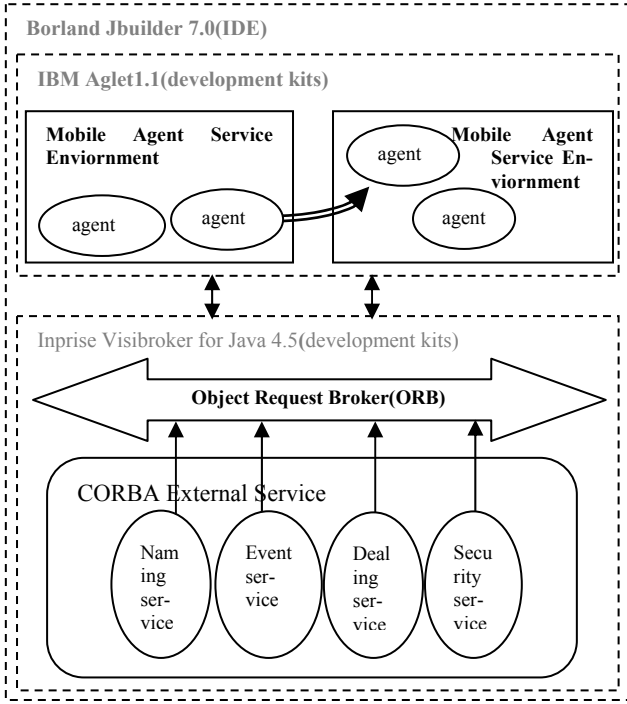


Fig. 2. MA\_CORBA prototype system

AgletContext” is in charge of tasks of agent system, and one agent system is defaulted as one place. There are some MA\_CORBA objects in the agent system, and they may leave the starting system and transfer to another system to deal with the appointed tasks.

### 3.2 MA\_CORBA Performance Evaluation

For validating MA\_CORBA system’s performance, we designed an instance of applying MA\_CORBA system to solve application server’s loading parallel problem in group environment. The whole test is executed in our lab’s local network composed of four host computers: Client<sub>A</sub> (CORBA client), Server<sub>B</sub> (MA\_CORBA server), and Server<sub>C</sub> and Server<sub>D</sub> configured the same as server<sub>B</sub>. The process of test is as follows: Client<sub>A</sub> sends request<sub>1</sub> to Server<sub>B</sub>. After Server<sub>B</sub> receives request<sub>1</sub>, it finds overloading at present through loading monitor, then packets the service object dealing with request<sub>1</sub> in CORBA environment to movable MA\_CORBA object, and then transfers the object to Server<sub>C</sub>. After a while, because Server<sub>C</sub> also runs in overloading state, it transfers the object to Server<sub>D</sub>. Finally, MA\_CORBA object runs in Server<sub>D</sub> properly and finishes the task completely, and sends the result back to Server<sub>B</sub> which sends the result back to Client<sub>A</sub>. The test adopts two methods to solve the problem of sending back the result: 1) Server<sub>D</sub> sends back Message (“any” type defined in CORBA

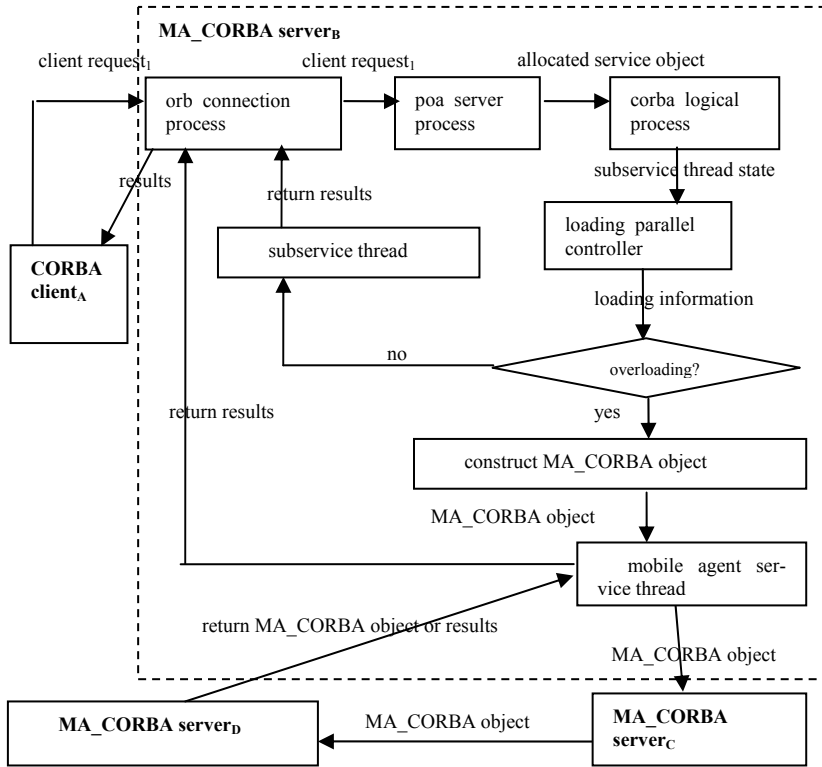


Fig 3. The application MA\_CORBA system in loading parallel

specification) object which saves the result to Server<sub>B</sub> through communication service provided by the mobile agent environment; 2) Server<sub>D</sub> transfers the MA\_CORBA object back to Server<sub>B</sub>, and Server<sub>B</sub> calls directly and locally the method provided by the MA\_CORBA object to acquire the result. The whole system’s data flow chart is shown in Figure.3:

- **CORBA logical thread:** It manages sub-service threads, including sub-service threads’ quantity, lifecycle, running state, etc.
- **Loading parallel controller:** It is application server’s loading control center, and monitors the local server and other servers’ loading state, and make the decision of transferring loading in terms of enlisting arithmetic, and finally mobile agent service thread implements this decision.
- **Constructing MA\_CORBA object:** This process is in charge of encapsulating objects or methods in CORBA server environment through class interfaces of IBM’s Aglet (mobile agent development platform) [9]. A simple instance is shown in Figure 4.

Boldfaced part in Figure 4 represents the codes which are added after a CORBA service class is packeted. These codes involve one base class and three functions. Aglet class is all MA\_CORBA classes’ base class, and all MA\_CORBA classes

must inherit Aglet class so as to be instantiated into movable MA\_CORBA objects. OnCreation() method resembles Applet class' init() method, and a MA\_CORBA class may be initialized into MA\_CORBA object if onCreation() method is overloaded. OnCreation() method can be only invoked once during MA\_CORBA object's lifecycle. When MA\_CORBA object arrives at a new host computer, onArrival() method will be invoked to initialize MA\_CORBA object. Once onArrival() method is executed, run() method is invoked so that MA\_CORBA object runs again in a new host computer. Run() method is MA\_CORBA class' executing thread's entrance point, and it deals with the material tasks allocated to MA\_CORBA objects [7].

- **Mobile agent service thread:** It provides MA\_CORBA objects' mobile agent service environment. It is in charge of receiving, sending and maintaining MA\_CORBA objects' running environment. For solving the problem of transparent location, a MA\_CORBA object's local proxy is created in mobile agent environment. MA\_CORBA object's transparent location is implemented through this proxy in spite of MA\_CORBA object's moving to any server [8].

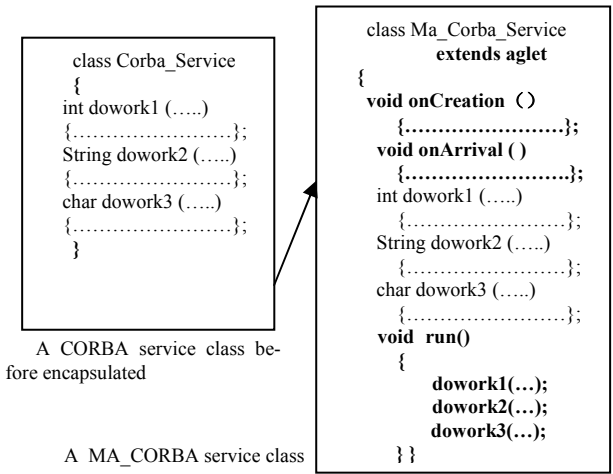


Fig. 4. The MA\_CORBA service class

The results of the experiment show that the MA\_CORBA prototype system operates smoothly except occasional vibration during the transfer of heavy loading. When a server's loading is heavy the server can implement loading transfer smoothly and return results to the clients, and when a server's loading is light, the server can transfer some loading from other over-loading servers automatically so as to improve the global performance of the group servers. Compared with the traditional loading parallel method, MA\_CORBA system not only transfers a segment of code but also transfers an active object including its state, which is very important to some long session users because they need the current and consistent states of MA\_CORBA objects to decide how to deal with the next request. In addition, in case of invalidation and overloading of the target server, MA\_CORBA object may adjust itself in terms of current environment and transfer automatically to another server.

## 4 Conclusion

On the basis of making full use of CORBA existing mechanisms, this paper studies especially the mobile agent system architecture in terms of OMG MASIF, and gives a mobile agent based CORBA server architecture and its prototype system which realizes the integration of CORBA and mobile agents in the common facilities layer of CORBA. Finally the prototype system is tested in loading balance in a distributed computing environment. MA\_CORBA architecture combines the major advantages of mobile agents and CORBA, and makes up shortcomings of each other. In MA\_CORBA system, CORBA object can move flexibly, and the movement does not damage CORBA object's transparency and persistence of naming mechanism. The objects in MA\_CORBA architecture is fully compatible to original CORBA objects.

The architecture has fine application prospect, and especially will play an important role in wireless network, mobile computing, initiative network, etc.

This paper only presents some results of our ongoing research work and a prototype system. A number of issues need to be further investigated, including system consistency, concurrency, openness, practicability, and compatibility.

## Acknowledgements

The work presented in this paper is partially supported by the National "863" Key Basic Research Development Planning Project (2001AA135180).

## References

1. Kotz D. and Gray, R.S.: Mobile agents and the Future of the Internet. *ACM Operating Systems Review*, 33(3) (1999) 7-13
2. Wang Y., Dispatching Multiple Mobile Agents in Parallel for Visiting E-Shops, *Proc. of 3rd International Conference on Mobile Data Management (MDM2002)*, IEEE Computer Society Press, Singapore, 2002) 61-68
3. Jennings, N.R., Sycara, K., and Wooldridge, M.: A Roadmap of agent Research and Development. *Autonomous agents and Multi-agent Systems*, 1 (1998) 275-306
4. Clements, P.C. and Northrop, L.M.: Software architecture: an executive overview. Carnegie Melon University, Technical Report CMU/ SEI- 96-TR- 003, ESC- TR- 976- 003 (1996)
5. Yang K., Guo X., and Liu D.Y.: Security in Mobile agent Systems: Problems and Approaches. *Journal of ACM Operating System Review*, 34(1) (2000) 21-28
6. Object Management Group. The common object request broker: Architecture and specification, <http://www.omg.org/corba>
7. The IBM Aglets workbench, <http://www.trl.ibm.co.jp/aglets/> (1998)
8. Gokhale, A. and Schmidt, D.C.: The Performance of the CORBA Dynamic Invocation Interface and Dynamic Skeleton Interface over High-speed ATM Networks. *Proceedings of GLOBECOM'96*, London, England, IEEE, (1996) 50~56.
9. Lang, DB., Oshima, M, Mitsum, O.: *Programming and Deploying Mobile Agents with Aglets*. Addison-Wesley, (1998)
10. Wong, D., Paciorek, N., and Moore, D.: Java-Based mobile agents. *Communications of the ACM*, 42(3) (1999) 92-102

# A Multi-agent Based Method for Handling Exceptions in Computer Supported Cooperative Design

Feng Tian, Renhou Li, M.D. Abdulrahman, and Jincheng Zhang

Systems Engineering Institute, Xi'an JiaoTong University, Xi'an, ShaanXi Province,  
P.R.China, 710049

ftian@sei.xjtu.edu.cn, Rhli@mail.xjtu.edu.cn,  
muhadmud@yahoo.com, jchZhang@sei.xjtu.edu.cn

**Abstract.** Focusing on exceptions that occur frequently in Computer Supported Cooperative Design (CSCD), the definition and classification of the exceptions are presented. According to expected and unexpected characteristics of exceptions, three methods are proposed: (1) expected exceptions are automatically dealt with by using Agent Message Event Rule (AMER) during the execution of collaboration processes; (2) document related exceptions were dealt with by adopting Document Tracking Log; (3) the cause of unexpected collaboration exceptions (UCE) is analyzed by using the algorithm of similarity-matching based on knowledge for mining exception cases in order to get the solution of similar exceptions. A prototype system, CoopDesigner, based on the proposed methods, is presented at the end of the paper.

## 1 Introduction

Computer Supported Cooperative Design (CSCD) is one of the major applications of Computer Supported Cooperative Work (CSCW) in the engineering design and one of the core technologies of concurrent engineering which provides an environment for collaborative product developments in an enterprise.

Presently, the workflow technology is adopted by many collaboration process management systems [1], but research on the exception handling and evolution of workflow systems are still under the way [2]. Chiu et al. [3] solved the problem of exception handling by using object-oriented workflow management systems combined with web technology; PROSYT [4] addressed inconsistencies and deviations in general process support systems; Klein [5] adopted a knowledge-based approach to handling exceptions in workflow management systems (WFMS) with emphasizing on agent management. Incorporating with the notion of the concept hierarchy Hwang [6] mined exception instances to facilitate unexpected workflow exceptions handling.

Collaborative Design is a method that is group-oriented, and has its own characteristics, such as group decision-making, document-centered, time-critical, knowledge centered and multi-disciplines. Since the requirements of handling exceptions and evolution in CSCD cannot be completely met by traditional workflow systems, a multi-agent based method to monitor and deal with collaboration exceptions is proposed in the paper.

Based on a brief summary of relative works [2-5], the paper is organized as follows: The exceptions are defined and classified in section 2. Combining multi-agents

with knowledge, a method to detect and handle the expected exceptions is presented in Section 3, and two methods for handling unexpected exceptions are described in Section 4. Finally, a prototype system called CoopDesigner, constructed on the proposed methods is presented in Section 5.

## 2 Exceptions in CSCD

### 2.1 Definitions of Exceptions in CSCD

We define an exception in CSCD as any deviation (such as resource conflicts, overdue etc.) from a normal execution or from ideal design results comparing with the required objectives of collaborative design. To make this clearer, let's take a detailed design of a collaborative construction design process for an example. Assuming that the whole collaborative process can be divided into three phases: 1) Task Decomposition, a task is decomposed into subtasks in the following sequence: architecture design; structural design; detail design for water and electrical supplies and distribution services, etc, and finally plan making in which a general task execution procedure including task precedence, hierarchy and timeliness is determined. The whole process consists of a series of declarations of detailed information and duration of tasks. 2) Task Assignment, the decomposed tasks are assigned to participating teams. 3) Task Execution.

Any deviation from the procedure above can result in bad influences on the collaborative design process, and lead to exception(s). The phenomena, such as overdue of structural design activities, poor quality of design results of an electrical design team, lack of human resource in a design team, etc, are regarded as exceptions.

### 2.2 Taxonomy of Exceptions and Related Concepts

As stated above, collaborative design is divided into three phases: collaborative task decomposition (CTD), collaborative task assignment (CTA) and collaboration task Execution (CTE). Correspondingly, exceptions that occurred in each phase are defined as: collaborative task decomposition exceptions, collaborative task assignment exceptions and collaborative task execution exceptions respectively. All the exceptions usually are referred to collaboration exceptions (CE) which is the basis of classification of collaboration exceptions (see in section 2.3).

According to whether exceptions are detected by using an agent's knowledge or a rule reference, they can be classified as expected collaboration exceptions (ECE) and unexpected collaboration exceptions (UCE) respectively, from which incur different method selection of exception handling.

### 2.3 Events, Rules and Exceptions

#### 2.3.1 Message, Events and Type of Exceptions

In agent technology based systems, the corresponding events are activated depending on the message transmission between agents. According to three phases of collaborative design, we can define three generic message types: collaborative task decomposition message, collaborative task assignment message and collaborative task execution



message. Normally, a message with a specific event type is sent to a relevant agent. When this agent processes the event and encounters an exception, exception handling agent (EHA) will be invoked to resolve it. This mechanism associates exception handling with agents' messages and events, and gives a basis that supports the online exception detection and handling. A hierarchical structure and taxonomy of a basic exception type is shown in Fig. 1. The exception taxonomy can be successively subdivided according to the phase and related events.

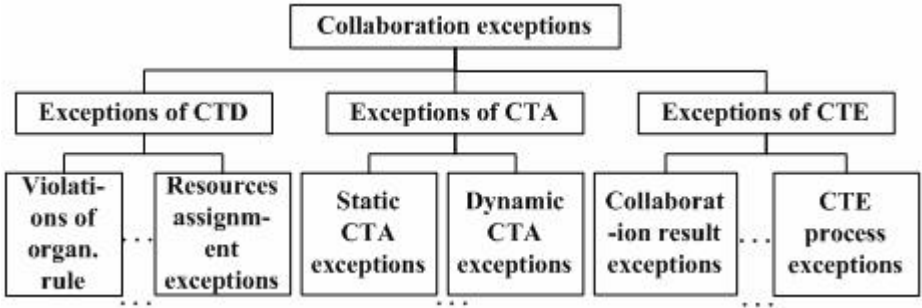


Fig. 1. Hierarchy of collaboration exceptions

### 2.3.2 AMER

In order to represent and process coordination constraints (objectives, organization/unit rules, policies, etc), collaborative task constraints (OR-Split structure under execution, etc), and meta-level task constraints (deadline, limitation of time interval, task status), an agent-message-event-rule (AMER) is proposed by combining agent communication language (ACL) with the message-event mechanism. AMER consists of following parts: agent-type, agent-event-type and rule sets. Agent-type describes the type of an agent which sends the message with an event; agent-event-type describes the event type of the message sent by the agent, which includes collaborative task management events (task creation, task modification, task deletion, etc), collaborative task execution events (start, finish, suspend, terminate, stop, resume and failure, etc), and events related to exceptions (knowledge conflict, loss during submission and distribution of document resources, overdue, data operation exceptions, route exceptions, task activation exceptions, etc). Rule sets consist of several production rules. Different agent event types are processed by different rule sets. Each production rule is formed by antecedent sets and consequent sets, the former normally are preconditions for executing, while the latter usually contains some specified operations or algorithms. The type of rule sets can also be classified as user defined type which is defined by users and stored in a rule base, and system defined type which is encoded in the algorithm of agents.

## 3 Handling Excepted Collaboration Exceptions

In our system, when an agent received a message with a specific event from others, the event is assessed and mapped to related rules. If exceptions are detectable and

predictable they can be caught and resolved by the predefined rules of AMER, which can improve the ability of exception location, control and automatic recovery during runtime.

### 3.1 Detecting Expected Collaboration Exceptions

The detection of the causes of ECE which includes matching the antecedent sets of ECE to the exception handling rules and inferring them according to the agent-type and agent-event-type of agents depends on operations of monitoring critical information conformed to the format of AMER. This process is activated online after a captured event is matched and confirmed by an agent. For a rule of exception handling, the antecedent sets describe the critical data, information and status of exceptions related to a collaborative task and several algorithms, and the consequent sets often specifies the exception handling mode, actions and some related operations.

Program 1 shows a basic format for a ‘finished-task’ message of AMER, which asked collaborative task executor (CTER) to evaluate an instance, Task<sub>j</sub>, and process a flow (see Program 2), calling for all constraints and rules after the instance of a collaborative task is finished and CTER receives a ‘finished-task’ event. In Program 2, IsTaskFinished() checks the system defined rules. Requestuserdefinedconstraints() checks the user defined rules. RequestRouteAgent() processes the precedence and hierarchic relationship among decomposed tasks and all basic control structures of workflow with the help of route agent (RA). RequestTaskAssignedAgent() asks Task Assigned agent (TAA) to process the dynamic resource allocation of the tasks that should be started. In the process, different agents are called one after the other and asked to detect different expected exceptions. Once an exception is detected, it should be resolved by EHA

Program 1. A AMER format of requesting CTER to process a ‘finished-task’ event

```
Primitive:      Request
Sender   :      TianFeng
Receiver  :      CoopTaskExecutor
Reply-with:     NULL
Reply-to  :      NULL
Language  :      OntoCKML
Businessstyle: Collaborative Design
TaskStyle:      Evaluation of working drawing
Content {
  Agent-Type-k: Task-Client
  Agent-Event-Typei: FinishedTask
                  {Content: Taski}
}
```

Program 2. A concise flow of execution when CTER received a ‘finished-task’ event

```
Event.FinishedTask(Taski) {
  IF IsTaskFinished()=False Then
    {RequestExceptionhandlingAgent(); Exit;}
  ELSE IF requestuserdefinedconstraints()=Error Then
    {RequestExceptionhandlingAgent(); Exit;}
  ELSE{
    IF RequestRouteAgent()=Error Then
      {RequestExceptionhandlingAgent(); Exit;}
    IF RequestTaskAssignedAgent()=Error Then
      {RequestExceptionhandlingAgent(); Exit;}
  } }
```

### 3.2 Handling Expected Collaboration Exceptions

Generally, once an agent has detected an exception during processing an event, an exception message will be sent to EHA which resolves the message and executes related operations according to correspondent AMER. Taking an example, if there is a need to monitor overdue of a task execution, a rule can be defined as in Program 3. Once CTER captures the exception, it sends a message to EHA which resolves the event type of the message and operates on it according to the rule's consequent to activate a time management task. For another example, the control structure for handling exceptions in coordination policies mostly adopts feedback structure and OR-Split structure (see Fig. 2). This needs the cooperation of CTER, route agent and EHA, such as combined with the Rulei of feedback in Fig. 2 and Program.3, the Taskj will restart if the result of the Taski is not available.

Program 3. Fragments of predefined rules and constraints correspondent to different events

```

Event.FinishedTask (Taski)
Begin
  ...
  UserdefinedConstraintk:
    IF (Task.Name=Taski and Task.status=Finished and
        Task.Ratioofduration >= 120%)
    Then (Action = TRUE) and (Task.Name=Timemanagement
        and Task.Status=Start)
  ...
End
Event.Feedback //Rulei
Begin
  IF (Task.Name=Taski and Task.TaskStyle=Audition and
      Task.Result = Rejected)
  Then (Action = TRUE) and (Task.Name=Taskj and
      Task.Status=Start)
End
    
```

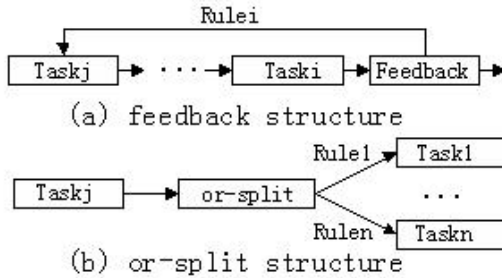


Fig. 2. Collaboration processes with rules

## 4 Handling Unexpected Collaboration Exceptions

The causes of unexpected collaboration exceptions (UCE), such as design conflict etc., are usually much more difficult to be detected. This implies that the means of predicting and detecting the causes of UCE by software agent are not existed and can only be figured out by experts after occurring UCE. But expert experiences are

always individualistic in nature and cannot be easily shared. To achieve the goal of sharing of exception handling knowledge and handling collaboration exceptions, this system adopts: 1) building documents tracking log (DTL) in order to quickly recover from trivial exceptions; 2) building exceptions case base, in which the processes of handling UCE by experts are recorded for the purpose of sharing and reuse. At the same time, it can also collect instances for capturing multi-expert experiences in exception handling to prevent subjective judgments while obtaining results directly (See the details in Section 4.2.).

#### 4.1 Exception Handling based on Documents Tracing Log

Due to the document-centered characteristic of CSCD, we seek the originating task with exception cause(s) by using exception handling wizard agent (EHWA) to track DTL. Such a method can recover original task not only by backward but also by forward. The items in DTL include originating task, originating version, parent version and parent task. Shown as in Table 1, a task of ‘assembling bolt accessory’ results in a file of ‘M16 bolt’ with parent version ‘null’, which means that such a file is the originating document file; the task of ‘Assembling robot arm’ outputs a file of ‘M16 bolt’, being input to the task of ‘Assembling bolt accessory’. A task that any deviation of design results has occurred could be found by tracking these logs according to the relations in document distribution among tasks, and then either backward or forward recovery is chosen according to the severity of the deviation. Forward recovery is adopted when a deviation is trivial and has no influence on other collaborative tasks, while backward recovery is adopted when severe deviations occur and have associated influence on many completed tasks, in this case the influenced tasks should be restarted executing.

**Table 1.** A fragment of DTL

Filename	Originating task	Originating Version	Parent Task	Parent Version
M16 bolt	Assembling bolt accessory	233	NULL	NULL
M16 bolt	Assembling robot arm	256	Assembling bolt accessory	233

#### 4.2 Handling UCE with Similarity-Matching Based on Knowledge for Mining Exception Cases

To compensate for the lack of methods handling UCE and to efficiently use expert experiences, we propose a method based on similarity match in exception cases by using knowledge to deal with UCE.

##### 4.2.1 Exception Record

After EHWA acquired a UCE and can resolved it with the cooperation of experts, the result and procedure would be recorded and stored in structured and semi-structured formats according to production rules and used by a mining algorithm based on

exception knowledge (see in section 4.2.2). The detailed items of exception record are shown in Table 2.

**Table 2.** The details of exception record

Record Items	Description
ID	Identification Number
Task Name	Task name which occurred an exception
Result	Result of exception handling: Failure or Success
Causes of Failure	Specified the causes after handling an exception
Mode	Exception handling mode
Recovery	The mode of recovery
Rules of exception handling	Antecedent set comprise TaskStyle, OBJECT, attributes of object, etc, consequent set include the operation of exception handling
other	Such as status, executor and temporal attribute of exception handling, etc.

### 4.2.2 An Algorithm of Similarity-Matching Based on Knowledge for Mining Exception Cases

Program 4. A concise programcode of mining algorithm

```

program MiningAlgorithm (Output)
{The current exception attribute set A[1...n]
 in which some elements should be: A[1](TaskStyle)
 A[2](Agent-Event-type), others should be the same
 style as: A[3]=O[3].a[3](Object.attribute);
 Exception instance set P[1...m]
 Candidate instance set CP[1...l]
 K is a maximum number of the returned results}
var
  int i,j,k,matchcount;
begin
  matchcount=0;
  FOR i=1 to Size(P)
  begin
    IF P[i].TaskStyle=A[1] and
       p[i].Agent-Event-Type=A[2]
    begin
      FOR j=3 to Size(A)
      FOR k=1 to Size(p[i].Rule)
      IF IsEqual(A[j],p[i].O[k].a[k])= TRUE
        matchcount++;
      Insert_Sort (CP[],p[i]);
    end
  end
  return CP[1...k]
end.

```

Taking an example, assuming that we know a set of symptoms of an exception,  $A(a_1, a_2, \dots, a_n)$ , extracting from the user input characteristic sets of exceptions by EHWA, then an algorithm (see Program 4) is used. Take Table 3 for an example, if an attribute set (*electrical distribution working drawing, distributor box, size*) is acquired, two similar records 21 and 22 would be matched. Thus, when locating the distributor box, a specified maximum size should not be violated if it was inserted in a shear wall, otherwise, it should be changed to other walls. It leads to a solution of the current exception, ‘distributor box location’.

**Table 3.** A fragment of exception instances

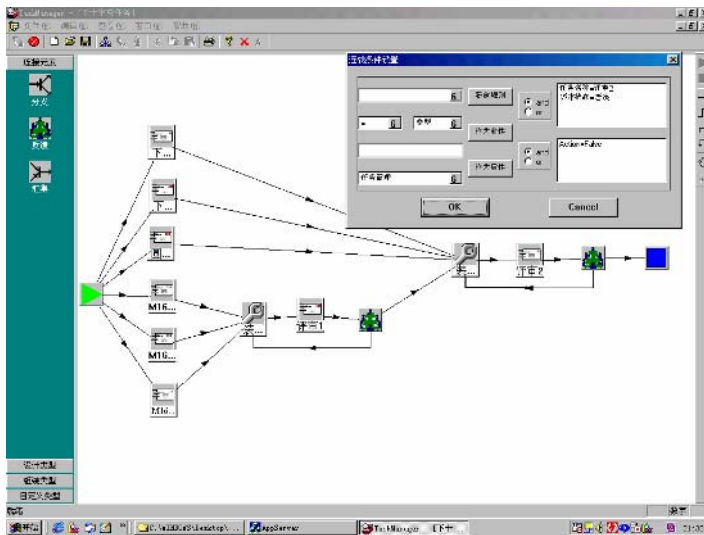
ID	...	Task Name	Exception Style	Result	Causes of Failure	Mode	Recovery
21	...	Electrical distribution working drawing	No available results	Failure	Hole size beyond the maximum allowed	Expert	Resume
22	...	Electrical distribution working drawing1	Modification failure	Success	Assembling bolt accessory	Expert	Resume

(Continued)

ID	Rules of exception handling
21	IF TaskStyle = Electrical distribution design and OBJECT=Distributor box and CAUSE = Smaller size THEN OPERATION = Enlarge hole
22	IF TaskStyle = Electrical distribution design and OBJECT=Distributor box and CAUSE = Size of Distributor box in a shear wall>4m <sup>2</sup> THEN OPERATION = Change the hole to near non shear wall

### 5 A Prototype System, CoopDesigner

A prototype system of collaborative design called CoopDesigner is designed and implemented. It consists of many subsystems including multimedia conference, tele-whiteboard, composite application sharing, production rule system, object-oriented knowledge base, multi-agent platform, collaboration organization management based on knowledge and collaboration task management [7], etc.



**Fig. 3.** A client interface of Collaboration Task Management

In the system, exceptions are detected and resolved online with the cooperation of clients of collaborative task manager (CTM), CTER, EHA, exception recovery agent (ERA) and EHWA, etc. CTM client realizes the collaborative task decomposition and

static collaborative task assignment, and has the ability to support the operations of the collaborative task evolution. Fig. 3 shows a task flow of designing a sub-accessory of a robot-arm, which includes 14 activities, and each of them is replaced with a box with a pixel map. Boxes with different image represents different kind of task-style, such as the one with a green triangle or a blue square represents a start activity or an end activity, the one with a pixel map of wrench means an assembling activity, etc.

After decomposing collaborative task and subsequent submitting to CTER, CTER controls the execution of the collaborative task. EHWA adopts methods of tracking DTL and algorithms of similarity-match based on knowledge for mining exception cases locate, resolve and recover exceptions with the cooperation of RA and ERA. Other instances of interfaces are omitted due to the length of this paper. The multi-agent system we implemented is modeled and analyzed by Object-Oriented Petri nets [8].

## 6 Conclusion

Combining multi-agent technology with workflow technology, this paper defined and classified the exceptions that frequently occur in CSCD. Three methods for exception handling are presented, in which according to expected and unexpected characteristics of collaboration exceptions and using routine rules, expert knowledge and experience, various agents divide the work and cooperate with each other to monitor the execution procedure on-line, detect exceptions and adjust the assignment of tasks in time (if there is an exception) in order to accomplish a common goal in an 'ideal' way.

## References

1. Shi, M. L.: Theory and application of CSCW, Publishing House of Electronic Industry, Beijing, (2000)
2. Alonso, G., Hagen, C., Mohan, C.: Enhancing the Fault tolerance of Workflow Management Systems, IEEE Transaction on Concurrency, July-September (2000) 74-81
3. Chiu, D.K.W., Li Q., Karlapalem, K.: Web interface-driven cooperative exception handling in ADOME workflow management system, Proceedings of the First International Conference of Web Information Systems Engineering, (2000) 174 -182
4. Hagen, C., Alonso, G.: Exception handling in workflow management systems, IEEE Transactions on Software Engineering, 26 (2000) 943 -958
5. Klein, M.: a Knowledge based approach to handling exceptions in workflow systems, Journal of Computer Supported Cooperative Works 9 (2000) 399-412
6. Hwang, San-Yih, Ho, S. F., Tang, J.: Mining Exception Instances to facilitate workflow exceptions handling, Proceedings of IEEE 6th International Conference on Database Systems for Advanced Applications (1999) 45 -52
7. Tian, F., Li, R. H.: Knowledge based approach for modeling collaborative processes, Proceedings of IEEE International Conference on Intelligent Information Technology-2002, Posts & Telecom Press, Beijing, (2002) 93-99
8. Tian, F., Li, R. H. He, B. Zhang, J. C.: Modeling and Analysis of Exception Handling based on Multi-Agent in CSCD, Journal of Xi'an JiaoTong University 38 (2004) 392-395

# CEJ – An Environment for Flexible Definition and Execution of Scientific Publication Processes

Daniel S. Schneider, Jano M. de Souza, Sergio P. Medeiros,  
and Geraldo B.Xexéo

COPPE/UFRJ - Federal University of Rio de Janeiro, RJ, Brazil  
{daniels, jano, palma, xexeo}@cos.ufrj.br

**Abstract.** An increasing number of sites including scientific electronic journals, digital libraries and collaboratories have been dedicated to scientific knowledge sharing and exchange in the recent years. Despite the apparent progress, in which traditional journals are migrating to Web and new forms of communication emerge, a closer look at these sites indicates that the scientific publishing model remains essentially unchanged. In this context was born the Configurable Electronic Journal (CEJ) project, a solution for generating configurable and extensible electronic journals. One of the main goals of this work is to apply open source technologies, including W3C Semantic Web technologies, in the design of an innovative environment for the definition of publication processes. CEJ environment is flexible to accommodate different styles of publications, suitable to be used in different processes and disciplines.

## 1 Introduction

The invention of writing, around 3500 B.C. in Mesopotamia, followed by the invention of paper by Ts'ai Lun, around 100 A.C in China, revolutionized the storage and transmission of knowledge. The next revolution in information technology came around 1440 A.C, when Gutenberg invented the printing press. The print technology reduced significantly the costs and time needed for reproducing manuscripts.

Now, we have entered the *Post-Gutenberg Galaxy*, “the fourth revolution in the means of production of knowledge” [8]. Compared to the previous revolutions, the computer technology revolution, besides reducing significantly the cost of storage and reproduction, provides entirely new capabilities: “Information can be transmitted instantly over long distances; knowledge which is stored electronically can be modified even after the time of publishing...” [1].

An increasing number of sites have been dedicated to scientific knowledge in the recent years. These sites have been called *scientific knowledge infrastructures* in the literature. Despite the rapid change and apparent progress due to the electronic medium and the Internet, a closer look at these sites indicates that the scientific publishing model remains essentially unchanged.

This paper presents the *Configurable Electronic Journal* (CEJ), a solution for generating configurable and extensible knowledge infrastructures. One of the main goals of this work is to apply open technologies of W3C Semantic Web [16], including the



eXtended Markup Language (XML), XPointer and annotation technologies, in the design of an innovative environment for the definition of publication processes. CEJ’s customizable workflow can accommodate different styles of publication, suitable to be used in different processes and disciplines. In this sense, it must be possible for the editorial board of the journal to configure which phases will constitute the publication process, which actors will be engaged in which phase and how, which form of discourse (open, closed etc) will be applied in the revision and discussion phases etc.

Other key goals of CEJ project include discovering new roles in the scientific publication process as well as experimenting with new collaborative forms of communication. To achieve the latter goal, we have designed an annotation component that allows documents to be augmented with comments and discussion threads over fine-grained elements of the document.

The rest of the paper is organized as follows. Section 2 presents the concept of *scientific knowledge infrastructures*, which has been the subject of this work. Section 3 compares the approach presented here with related work; Section 4 presents the CEJ environment and current experiments with the prototype, and Section 5 discusses the final considerations and future work.

## 2 Scientific Knowledge Infrastructures

Hars [1] defines the term online scientific knowledge infrastructure as “a socio-technical information system aimed at the creation, organization, storage and application of scientific knowledge. It is not only a technical system consisting of hardware, software, databases etc. It is also an organization system which consists of stakeholders, organizational structures and processes” [1].

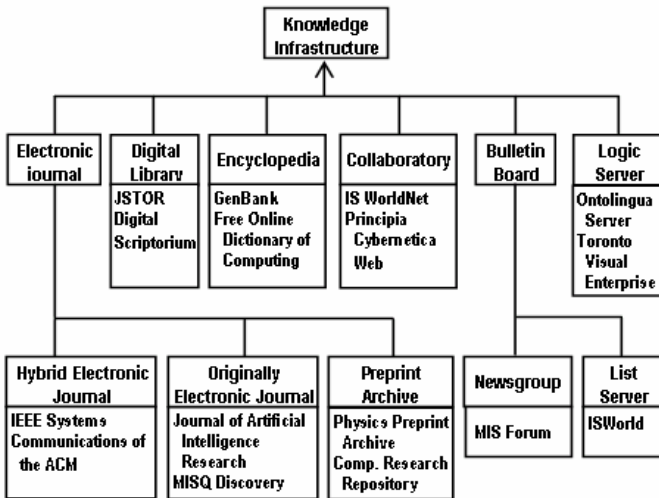


Fig. 1. Categories of knowledge infrastructures

Fig. 1 depicts a general characterization of these infrastructures, adapted from [1]. A great number of these sites can be categorized as “online” or electronic journals.

With the emerging of these infrastructures, we are moving towards a continuum of publication, and this requires a continuum of peer review, which will provide feedback to scholars about articles, and not just in the one-time binary journal decision process stage [11].

However, concerns about possible damage to the traditional review system are slowing down the pace of change. Despite the apparent progress, in which traditional scientific journals are migrating to Web, new forms of communication emerge, and some activities of the publication process are automated, the scientific paper-based publishing model remains unchanged, as mentioned before. It could be argued that some few electronic journals have been experimenting with new forms of academic discourse, and some significant improvements have been achieved. This is the case, for instance, of *BMJ*, a general medical journal, of *Psychology* in the cognitive sciences, and of the revolutionary model of publication in the *High Energy Physics*. However, if we consider the universe of 24,000 research journals published worldwide [18], these cases represent the exception, not the rule.

### 3 Background and Related Work

In this work, some knowledge networks were investigated. Particularly, we were interested in studying online electronic journals. The following models were investigated: the *British Medical Journal* (BMJ) [7], Stevan Harnad’s *Psychology* [14] and *Behavioral and Brain Sciences journal* (BBS) [15], the *PubMed Central* by NIH [2], the *Journal of Interactive Media in Education* (JIME) [4], Ginsparg’s Physics preprint archive [3], and finally, John Smith’s hypothetical “*deconstructed journal*” [6]. This research was preceded by an extensive theoretical analysis, with the aim to understand the traditional publication model, and also how the electronic medium and the Internet open new doors to improve the traditional system. For instance, the fact that articles are bundled into “issues” in traditional print journals is due to the economics of publishing.

Currently, we are aware of only a few systems that provide functionality similar to CEJ. JIME (The Journal of Interactive Media in Education) is a good initiative conducted by the Knowledge Media Institute, in UK, with the aim to redesign the traditional peer-review process. They have been designing a computer-mediated technology to support new kinds of discourse. One of the challenges for JIME’s work group is shifting reviewing from a closed process to an open process promoting constructive dialogue between participants [4].

A key aspect of their technical design is the integration between the document and the discourse. However, JIME and the known *D3E Environment* lack some potential good features. First, documents must be submitted in the HTML format, limited as a publication format, as we will discuss in section 4.2. Secondly, there’s no support for fine-grained annotations, allowing readers and reviewers to comment on a specific paragraph, figure or a sentence. Finally, the system does not support the definition of customizable publication processes. In fact, it does support one fixed publication lifecycle, only useful for disciplines in which softer criteria for acceptance/rejection

must be negotiated [4]. Therefore, it's hard to believe that JIME would be suitable for electronic journals in other disciplines like Physics and Computer Science.

## 4 The CEJ Project

In this section we detail the CEJ environment, starting from the system design, then discussing the support for collaboration and the system architecture, and finally presenting the current experiments.

### 4.1 Design Principles

This work improves JIME's efforts to redefine the scholarly publication process in three main points: (1) content is separated from presentation, which is an important requirement in that it allows the same article to be published in more than one journal, with different presentations. Multiple virtual documents can also be created from the same original document, suitable for use by different audiences in the same journal; (2) documents can be augmented with comments and discussion threads over fine-grained elements of the document, allowing readers and reviewers to comment on a specific paragraph, figure or a sentence; and (3) publication processes can be defined in a flexible way, through CEJ's workflow engine.

In our object-oriented data model, a *journal* can be associated with any number of *process templates* and may run several *process instances*. A *user* can perform a number of *roles* in a given journal, and each role has its own set of access permissions, defined by the editor. *Articles* are published in a journal and have a *type*, which may be an *original article*, a *survey*, a *review* etc. *Documents* are associated to articles, and also have a *type* (*proposal*, *pre-publication* etc). An *annotation* has a *type* (e.g. *comment*, *question*), is created by a user and either annotates a document or responds to another annotation. Annotation granularity may range from a single word to the whole document.

At the heart of the CEJ system is an object-oriented workflow engine developed for the Web. The flexibility of this component allows easy configuration of workflow templates. Typically, any workflow template will be a combination of some "elementary" tasks, pre-defined in CEJ environment. However, it is easy to extend these classes (or to create new ones from scratch) with the aid of a Java developer.

### 4.2 Document Format

Among emerging electronic journals, scientific articles are usually submitted in PDF (Portable Document Format), PS (PostScript) or HTML (Hypertext Markup Language) formats. HTML has the advantage that it does not require any software beyond a web browser. Most word processing tools supports the translation to HTML. Besides that, the language provides the standard formatting and font size. On the other hand, the most common format is PDF, which requires the Adobe proprietary software to edit the document, but provides the advantage of preserving the look and feel of the original document (e.g. layout, size, etc.). PDF documents are viewable over the web from within a web browser, similar to HTML, but require a separate plug-in to be read. In addition to this, the format is not integrated with Web technologies like

W3C DOM and JavaScript. PostScript is similar to PDF, but cannot be viewed within a standard web browser.

Over the last years, the work in the area of Semantic Web has shown that the XML technology emerged as the Internet electronic document standard. The Semantic Web aims at machine agents that thrive on explicitly specified semantics of content in order to search, filter, condense, or negotiate knowledge for their human users [19]. Choosing the XML standard means that every article in the journal repository can be broken up into its fundamental parts - authors, affiliations, sections, references etc. Additionally, in contrast to HTML, XML offers the ability to separate content from presentation.

In CEJ project, we have chosen the XML standard as the file format for article submission. In CEJ first prototype, we decided to use *DocBook*, an open source XML format for publication of books and articles. In similar way as in PubMed Central journal [2], CEJ provides a *validator module* to check for completeness and syntactical correctness. This API can be used, for instance, to guarantee that every article in a specific journal contains at least one author, that an abstract for the article is required, and that the article body should not exceed 5000 words in length.

### 4.3 Support for Collaboration

The nature of academic discourse is changing, becoming more interactive, collaborative and interdisciplinary. Another key goal of CEJ project is experimenting with new collaborative forms of communication in the publication process. To achieve this goal, we have designed an annotation component that allows documents to be augmented with comments and discussion threads over fine-grained elements of the XML document.

The CEJ annotation module uses the *proxy-based* approach, in which annotations are stored and merged with the Web document by a proxy server. *Browser-based* approaches, in which a specific browser is modified in order to merge the document with annotations, were discarded to facilitate widespread applicability. The merging of a document with associated annotations is done on the server side, producing a "virtual document" in the client browser. CEJ provides an interface for querying documents with associated annotations, as part of its object-oriented API. This allows readers to see different virtual documents based on the same article.

In CEJ system, the visualization of annotations can be configured in a variety of ways. For instance, an annotation can be visualized either as a hint, in a separated window, or embedded in the virtual document.

Annotations are stored in the database, and are accessible through the object-oriented API. Any part of the XML document can be annotated: the whole document, an abstract, a section, a paragraph, a figure, a string range of a paragraph etc. We use *XPointer* for locating the annotations in the annotated document.

### 4.4 CEJ Architecture

Since the beginnings of this project, we aimed to make CEJ a flexible and robust system. In order to achieve this goal, we have chosen Struts, an open source framework for Java Web applications, as the base framework for developing the CEJ

system. The framework technique is a software engineering technique for producing a reusable, “semi-complete” architecture that can be specialized to produce custom applications [12]

Struts [13] is based on the Model-View-Controller (MVC) design pattern, which separates the application object (model) from the way it is represented to the user (view), and from the way in which the user controls it (controller). Another advantage with the MVC pattern is providing a means of organizing systems that support multiple presentations of the same information. Fig. 2 shows CEJ’s system architecture:

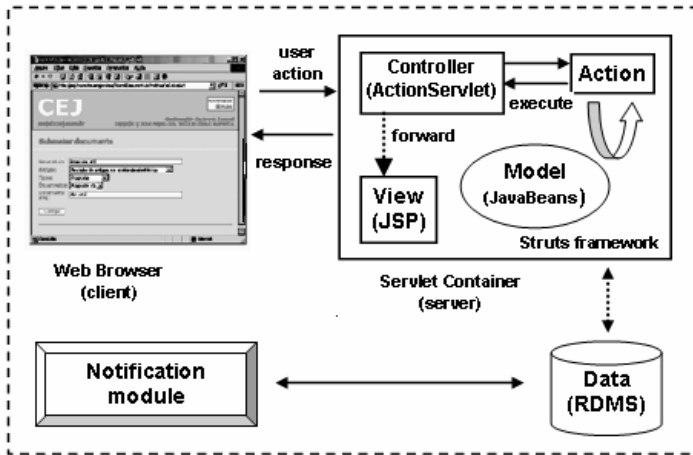


Fig. 2. CEJ architecture

Since the system is based on Web technology, besides a standard Web Browser, no special software is needed on the client-side. The client request is first intercepted by a Java servlet, referred to as a *controller servlet*. This servlet handles the initial processing of the request and determines which view page to forward to the user. The model classes in CEJ system are a set of simple Java objects. As a data store, we are currently using an instance of the MySQL relational database, but any other DBMS system (e.g. PostgreSQL, Oracle, ...) could be plugged, provided that it supports a JDBC driver.

The views within the architecture consist of JSP pages, augmented with Struts tags. Most dynamic content is generated in the server, but some business in CEJ system required client-side JavaScript, especially in the implementation of the annotation system. Finally, the system architecture also includes a notification module.

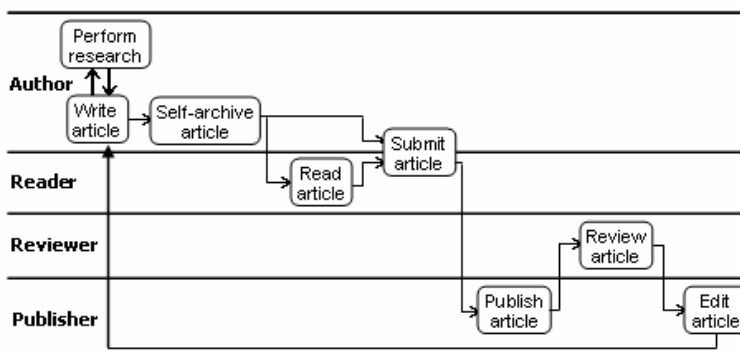
#### 4.5 Experiments

Since the early days of this project, we have been interested in experimenting with a few scenarios of publication. Particularly, we will discuss here two models of publication which have been set up in the CEJ portal:

1. A continuous flow journal with instant publication.
2. An invitational journal with a proposal phase.

*Model 1: A continuous flow journal with instant publication*

In this model, authors perform research, create and edit the article, and then self-archive the article on their personal servers. Once the article is available on the Web, readers can read it. As shown in Fig. 3, the article submission can be performed either by the author, or by the reader! The “submission” role could also be performed by the journal’s editor, searching the Web for quality articles. After successful submission, the journal incorporates the article through a link (URL). Provided that none of the journals holds its physical control, an article can be published in several journals at the same time.



**Fig. 3.** The continuous flow model with instant publication

Note that publication in the journal occurs immediately after submission, and is performed before the peer-review phase. The cycle ends when the editor (or publisher) performs copy-editing, and the article may be returned for update by the author. The continuous flow model is shown in Fig. 3.

To implement this model in CEJ, four tasks were required to be set up: a submission, a revision, an approval and an update phase. All of them are defined as elementary phases in CEJ environment. The submission task was configured to accept only URLs. In the revision phase, the journal was configured to support open peer-review, and CEJ annotation system was set up to allow a variety of contributions and types of annotation. The approval task allows editors and reviewers to select the contributions from the review phase and to incorporate them into a virtual document. Finally, in the update task, a deadline can be set up for editing by the author.

There are many opportunities for collaboration in the instant publication process described here. Collaboration in the form of annotations, discussion threads and peer-commentary can begin as soon as an article is submitted and instantly published, and may be performed in parallel with the peer-review activity. Contributions may continue even after the review and update tasks.

Some considerations regarding “model 1” deserve to be mentioned here. First, self-archiving has been gathering force over a decade, and Ginsparg’s Physics preprint archive [3] is an example of a successful implementation of this model. Secondly, in contrast with the traditional paper-based model, peer-review is performed after publi-

cation here, as we mentioned, reducing the publication cycle time significantly. This approach is based on the argument that, in the electronic environment, publishing is reversible. According to Hars [1], "the counter-argument that this might lead to a flood of low-quality articles incorrectly assumes that the reader is not able to distinguish between unreviewed and accepted articles". In addition, filtering technology can be used to selectively eliminate undesired articles. In CEJ system, stamps can be attached to documents in order to distinguish between reviewed and unreviewed articles. The interface also allows users to access only reviewed articles.

*Model 2: An invitational journal with a proposal phase*

In this second model, the editor invites an author to publish based on the reading of previous work by that author. The author writes a proposal, which is then reviewed and negotiated. A successful negotiation results in the writing and publication of a new article.

One of the key advantages of this approach is that almost every article written gets published. The disadvantage is that selection of authors may be arbitrary and there is no way an unknown author can get published [10].

In order to implement this model in CEJ, we have defined a "proposal phase", composed by a submission, a review and an update phase. A second submission phase was also required for the new article. Collaboration between authors and reviewers occurs in the negotiation phase, making use of CEJ annotation system, and may proceed with the new article after publication. The invitational journal model is shown in Fig. 4.

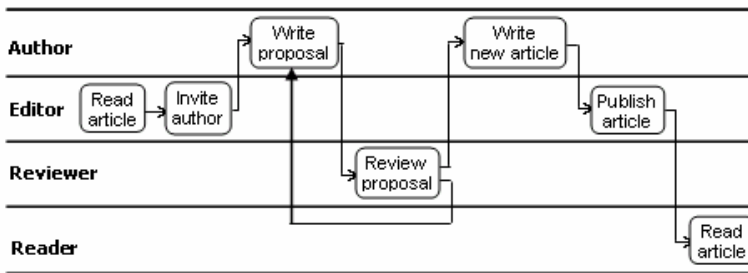


Fig. 4. An invitational journal with a proposal phase

Finally, it's important to mention that traditional journals are mainly targeted at the final stages of the research process, because of the inherent limitations of paper to support fruitful discussions [1]. The proposal phase fills a major gap in traditional journals by allowing authors and reviewers to directly cooperate in the early stages of the publication workflow. This cooperation can be useful for both publishers and authors. It helps the authors compose a paper that reaches out to the journal's audience. On the other hand, it can also help the identification of a good new research problem based on previous work.

## 5 Conclusions and Future Work

We believe that the current emerging of scientific knowledge infrastructures is only the first step towards redesigning the academic publishing model. The scientific publishing workflow will probably turn into a collaborative and interactive process, and peer-review will likely remain a part of it, but will possibly be open and supplemented by other mechanisms like peer-commentary. XML documents are likely to become the document type of choice in the scientific electronic environment.

In this paper we presented CEJ, an infrastructure for generating configurable and extensible electronic journals. The purpose of this project is to experiment with new collaborative forms of academic publishing, making use of open technologies of Semantic Web.

We are currently working on extended support for electronic conferences and investigating the possibility of supporting collaborative Web-based editing of documents, in the XML format. In the near future, we also plan to support: (1) versioning, taking advantage of the fact that documents are stored in the XML format; (2) more interactive forms of publication; (3) translation to PDF format using Apache FOP engine [17], another open source tool.

In the recent years, a worldwide movement to make free online access of scientific literature through institutional repositories has been gathering force, known as the "Open Archives Initiative" (OAI). One of the actions of this initiative is offering free software to institutions that wish to create OAI-compliant e-print archives. In the future, we aim to turn CEJ compatible with the OAI protocol and patterns.

We hope this work can contribute to experimenting with new emerging academic models, to discovering new roles and possibly new actors for the publication process, and also to opening new perspectives in studying knowledge infrastructures, in that it emphasizes the power of the electronic medium and the open, standards-based computer technology.

## References

1. Hars, A.: From Publishing to Knowledge Networks, Springer, Germany, (2003)
2. Kling, R., Fortuna, J., and King, A.: The real stakes of virtual publishing: The transformation of E-Biomed into PubMed Central, Indiana University, Center for Social Informatics, 2001.
3. Ginsparg, P.: First steps towards electronic research communication, *Computers in Physics*, 8(4) (1994) 390-396
4. Sumner, T., Shum, S. B.: Open Peer Review and Argumentation: Loosening the Paper Chains on Journals, *Ariadne*, (1996)
5. Harnad, S.: The self-archiving initiative, *Nature*, 410 (2001) 1024-1025
6. Smith, J. W. T.: The Deconstructed Journal - A new model for Academic Publishing, *Learning Publishing*, 12(2) (1999) 79-91
7. Smith, R.: Opening up BMJ peer review, *BMJ*, 318 (1999) 23-27
8. Harnad, S.: Post-Gutenberg Galaxy: The Fourth Revolution in the Means of Production of Knowledge, *Computer Conferencing: The Last Word*, (1991)
9. Odlyzko, A.: The rapid evolution of scholarly communication, *Proceedings of the 1999 PEAK conference*, (2001)



10. Stodolsky, D.S.: Consensus Journals: Invitational journals based upon peer consensus, *Psychology* 1(15) 1990.
11. Odlyzko, A.: Peer and non-peer review, *Peer Review in Health Sciences*, 2nd ed., F. Godlee and T. Jefferson, eds., BMJ Books, (2003) 309-311.
12. Fayad, M.E., Schmidt, D.C. and Johnson, R.E. : *Building Application Frameworks – Object-Oriented Foundations of Frameworks*, John Wiley & Sons, Inc., (1999)
13. The Apache Struts Web Application Framework: <http://jakarta.apache.org/struts/>
14. *Psychology*: <http://www.princeton.edu/~harnad/psych.html>
15. *Behaviour and Brain Sciences*: <http://www.bbsonline.org>
16. The Semantic Web: <http://www.w3.org/2001/sw>
17. The Apache XML FOP Project: <http://xml.apache.org/fop>
18. Harnad, S: Knowledge freely given should be freely available, *Montreal Gazette*, (2004)
19. Handschuh, S., Staab, S.: *Annotation for the Semantic Web*, IOS Press, (2003)

# Methodology of Integrated Knowledge Management in Lifecycle of Product Development Process and Its Implementation

Peisi Zhong<sup>1</sup>, Dazhi Liu<sup>2</sup>, Mei Liu<sup>1</sup>, Shuhui Ding<sup>1</sup>, and Zhaoyang Sun<sup>1</sup>

<sup>1</sup> Advanced Manufacturing Technology Center,  
Shandong University of Science and Technology, Qingdao, 266510, P.R. China  
{pszhong, liumei, shding, zysun}@sdust.edu.cn

<sup>2</sup> College of Mechanical Engineering, Shandong University, Jinan, 250061, P.R. China  
dzliu@sdu.edu.cn

**Abstract.** This paper first provides a literature review of product development and knowledge management including product development process, design history and domain knowledge. It then presents a method for knowledge-based multi-view process modeling including process implementation, process monitoring, and knowledge management. The integrated framework and hierarchical model of the design history is built. The relationship among process history, design intent and domain knowledge is analyzed, and the method for representation and acquisition of process history, design intent and domain knowledge is presented. The multi-agent based architecture of the knowledge-based product development process management (PDPM) system and its functional modules are described, and the architecture of integrated knowledge management (IKM) system based on PDPM is set up and developed with a B/C/S (Browser/Client/Server) structure. The system is used successfully during the lifecycle of a new type of railway rolling stock development.

## 1 Introduction

Concurrent engineering (CE) is a systematic approach to integrate concurrent design and its related processes. Its objective is to shorten the product development cycle, improve the product quality and reduce the product cost. More and more research work for concurrent engineering is focused on the product development process modeling (PDPM), implementation, design history, team organization, logistics, the key characteristics of product design, dynamic problems in process implementation, process management and so on.

PDPM is one of the key enabled technologies for the implementation of concurrent engineering including process modeling, process implementation, process monitoring, process analysis and optimization, process improvement and reengineering, process simulation and so on. In most cases of the product development process, the experience of the product development is acquired passively rather than actively. Because the designers and manager of the project are busy in developing the product, and have no time to build the knowledge base for product or improve the development process.

The main purpose of this study on integrated knowledge management (IKM) is to explore the theory and methods on process management, history management and domain knowledge management in the product development process, that is, to integrate the knowledge management into the lifecycle of the product development process including mainly the acquisition and reuse of process history, design intent, and domain knowledge.

## 2 Literature Review

The study on traditional product development originated from 1960's and became a active topic at the end of 1980's and the beginning of 1990's. With the embedded study on CE, the cooperative product development, integrated supporting environment for concurrent design and intelligent decision support system become the current hotspot. The integrated product development team for cooperative product development is made up of designers who are in different sites and connected by network, as if they were in one place. Thus it can reduce the development cost and improve product quality by fully using the software and hardware resources of CAD/CAM (computer aided design / computer aided manufacturing) and all kinds of knowledge and experience of related designers [1][2][3]. How to acquire and reuse the knowledge during the development process becomes a research hotspot now.

The IKM system is a complex multi-object man-machine cooperating environment involving concurrent product development process, system structure of networked knowledge management, design history, domain knowledge representation and acquisition, decision support, supporting environment etc.

### 2.1 Product Development Process

Product development process is the basis of the research on IKM. The system of product development process management includes process modeling, process analysis, process improvement and process reengineering, process monitoring and conflict management [4]. According to different requirements and applied backgrounds, researchers proposed various kinds of process methods and technologies. Summing up all methods, there are IDEF method, structured analysis, Petri-net modeling method, real-time structured analyzing and process modeling method, process programming method (including rule-based method) and systematic dynamic method [5]. In these methods, some of modeling methods are supported by corresponding tools such as SADT/IDEF0 (structured analysis and design technique / Integration Definition for Function Modeling), Petri-net and so on, others are still at the stage of academic research and lack of supporting tools [6][7]. All of them and their supporting tools only focus on one or two views of many views of product development process, especially not support the knowledge view including process history, design intent and domain knowledge in the lifecycle of product development process [8][9]. That is to say, they only solve part of all problems in PDPM and are lack of full-scale description of the process. And there is not a standard, acceptable and acknowledged method which is necessary. In fact, if there is no any acceptable evaluating standard, it is difficult to evaluate a method of product development process modeling and management.

PDPM can assort with product development activities, monitor and validate product development process, build unified product process information model, set up the checking mechanism of product development process, establish the mechanism of constraint and conflict resolution among activities, provide decision support tool for product development process and design and so on.

## 2.2 Process History, Design Intent and Domain Knowledge

The research on IKM mainly focuses on the domain knowledge acquired during product development including design decision knowledge and process decision knowledge. The knowledge for design decision includes knowledge of requirement of market and users, design standard, design principle, design modification, manufacturing, assembly, cost, and quality. The knowledge for process decision includes information on manpower and resource, knowledge of process planning, and coordination. All the knowledge reflects the history of product development and plays an important role in designing high-quality products.

In the existing models for design history, only the data of product, process (including operation) and decision (including intent, principle and constraint) are the main contents, and the whole process from the initial definition to the final geometry and topology design can be backdated according to the decision, and the information of product, information of development process and their relationship can be retrospectively according to the process [10][11].

However, it is difficult to integrate design history, process model and product structure. In most researches, process development is considered as task flow, the process itself is not paid attention to, then how to integrate process, domain knowledge and numerical product model is not thought over.

How to integrate process knowledge and decision principle to geometry model and function model of product relies on more ordinary integrated product model [12]. As design is dynamic, how to represent the dynamic history information also needs the support of product model. How to represent efficiently domain knowledge is a complicated problem. Knowledge acquisition is lack of efficient development tools, at the same time, it is the first step and most difficult and important link to build a knowledge-based systems [13][14][15][16][17].

## 3 Multi-view Based Integrated Framework

As there is so much information in the lifecycle of product development process especially for complex products, it is difficult to describe all information in one view. According to the different information, it can be divided into many groups, each view describes a group of information and then all views are integrated together.

### 3.1 Basic Concepts

The integrated knowledge in lifecycle of product development process includes the following main parts:

(1) Process history is the history of the design tasks, including the steps of process implementation, the statuses of the implementing condition, the statuses of the flows,

the distribution of organization and resources, the schedule of the project, as well as the record of evaluation, and decision etc.

(2) Design intent is the sum of information about the design method and the decision causation, including the status changing process of the design information, the design steps and scene which caused the change, as well as design history information of design decision, the choice of design schemes, design intent and so on.

(3) Domain knowledge is the sum of the design principle, design method and design experience which exist in professional books, manuals, as well as in the brain of human.

### 3.2 Multi-view Based Process Model

The PDPM system is developed to coordinate activities in concurrent product development process, control and validate the product development process, set up a union information model for product and process, mechanism of examining and approving for product development, mechanism of constraint and conflict coordination among activities, and provide toolkits for PDPM system.

When modeling the product development process, different people such as managers, developers and engineers, may have different requirements for the process. It is necessary to describe the process in different aspects and form different views for the process. So the ideal method is to build an integrated model based on multi-view for product development process.

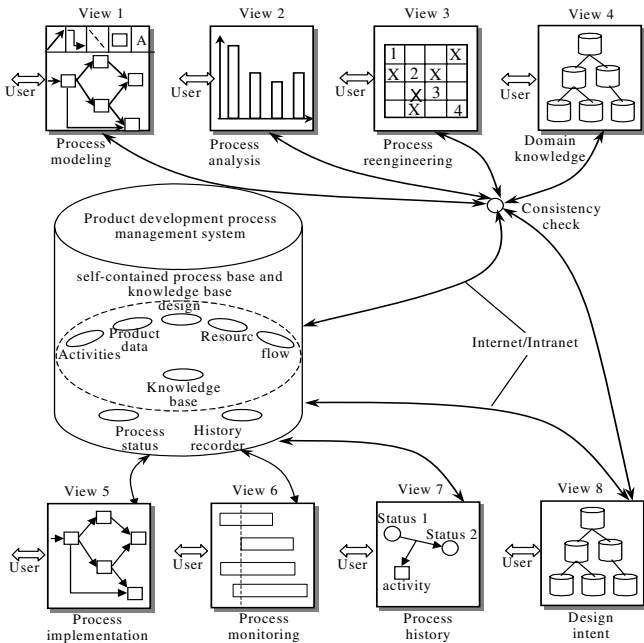


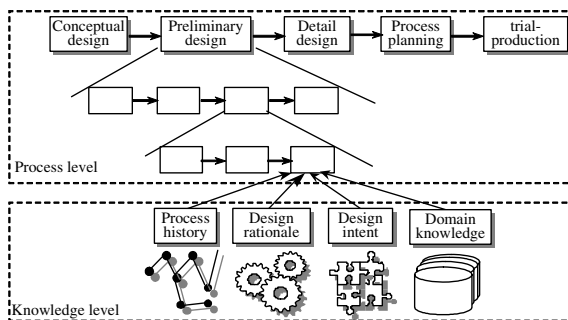
Fig. 1. The principle of multi-view based process modeling

Figure 1 describes the principle of the integrated process modeling based on multi-view. In the model, users can describe and use different parts of the process and different stages of the development process with different computers in different places. For example, View 1 represents that users can model the different parts of the process with a few computers and describe the activities, roles and resources. View 2 represents that users analyze and optimize the process model. View 3 represents that users improve and reengineer the process model. View 4 represents that users require or reuse the domain knowledge. The model built by a few users must be checked in consistency by process modeling tool so that a self-contained executable model can be set up for product development. The process model is implemented in actual project. The PDPM supporting tool can generate different views such as activity-network (in View 5), status transaction of key data (in View 6) and so on according to the demands of users. And users also can browse, implement and monitor the process in one view with any computer anywhere, for example, to distribute resources, to reengineer the process or analyze the process schedule using a design tool, to capture the process history in the development process (in View 7), to capture or reuse the design intent for product development process (in View 8) etc.

### 3.3 Integrated Knowledge Framework

The integrated knowledge base is the recorder and elucidation about the design object in the product development process. It incarnates all important information and professional knowledge in the process of product development lifecycle.

In general, the model of integrated knowledge involves a number of stages such as conceptual design, preliminary design, detail design, and process planning. In these stages, the design object changes from abstract to concrete form, and the design data are generated and perfected step by step. The interaction among the designers, design intent, design data, design tools, resources, and environment changes and forms the design pathway. The recorder to the thinking process of designers forms the design rationale. Figure 2 is the hierarchical model of integrated knowledge including process history, design intent, domain knowledge and so on.



**Fig. 2.** The hierarchical model of integrated knowledge

There are two levels in the model including process level and knowledge level with tree structure. In process level, the product development process is divided into

conceptual design, preliminary design, detailed design, process planning, trial-production and so on, and each sub-process can be decomposed further. In knowledge level, the knowledge is also divided into several sub-knowledge bases such as process history, design intent, and domain knowledge, and each sub-knowledge base can also be decomposed further. Every sub-knowledge base is corresponding to one sub-process in the model.

### 4 Representation and Acquisition of Integrated Knowledge

The design intent and domain knowledge can be acquired by knowledge acquisition tools and the correlative attributes of design are automatically acquired by the integrated framework, such as time, correlative activities, and roles etc. The process history mainly uses automated acquisition supported by computers except acquiring mechanism of process management similar to design intent. By integrating the process implementation and monitoring system, all events in the development process can be captured. And the module of process history will capture the events automatically, and save them in the process history base, as shown in Figure 3.

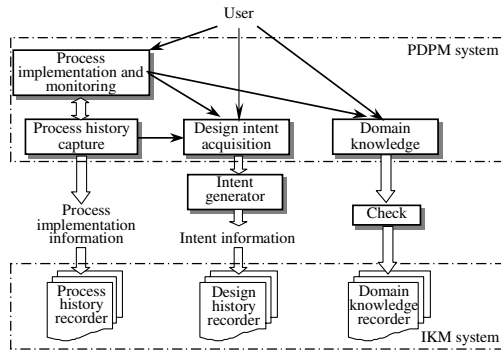


Fig. 3. The relationship of design history and process management

#### 4.1 Process History

Process history is the record of the actual product development process according to the technology management. The content of process history includes three parts:

- Part 1 is the process model of the special product development. It mainly includes process architecture, activities, descriptions of input and output flows, constraint condition and rules for process execution, status of flows, distribution of roles and resources. The information above can be acquired when process is modeled.
- Part 2 is the executing history of design tasks, including changes of development process, sequence and time of task execution, status change of process execution condition, as well as changes of activities and flows, dynamic usage of roles and resources, reason of design changes and iterative process, and dynamic calculation

of workload. In addition, public messages and special messages are provided in the environment of concurrent product development process.

- Part 3 includes coordination and decision of tasks or project and its principle, and content corresponding to the project, such as time and cost of designers training, method or technology involved in the project, the domain of developers, cost of evaluation, and time of document writing.

All the three parts are acquired automatically during the process execution as shown in Figure 3. The module of process history capture acquires the process implementation information from the module of process implementation and monitoring, and input all this information to the process history base.

## 4.2 Design Intent

The capture of design intent is a semi-automatic process with part of autonomic capability and part of participation of users. There are two input methods: (1) the system activates the intent capture module through the process monitoring module according to the property and result of tasks, and requires developers to input related information; (2) the developer may activate the intent capture module at any time according to his requirement, as shown in Figure 3.

Design intent uses the natural language representation. It supports the non-formalized input tools such as text editor, notebook, electronic panel, e-mail and multimedia device. Thus the developer can describe his ideas more clearly and freely, without being constrained by rules.

## 4.3 Domain Knowledge

The product design is a complex process and must be researched with the system engineering approach. The components have very strong independency. So the goal of product design can be decomposed into several sub-goals and each has stated independently. The relationship among sub-goals forms a public knowledge base (KB), and the knowledge for each sub-goal forms its own KB which can inherit the knowledge from the public KB.

The knowledge for each sub-goal can be concluded into a series of clusters of concepts, and each one can be decomposed into a series of attributes. The relationship among clusters is OR, and that among attributes is AND. The attributes include two kinds: type and scope. The 'type' reflects how to class the attributes and the 'scope' reflects the distribution of attribute values. In fact, the attributes can also be regarded as variables. Figure 4 shows the AND-OR tree based model of concepts.

On the other hand, the process of product design can be divided into a few sub-processes, such as conceptual design, preliminary design, detail design and process planning as shown in Figure 2. The different stages of product design is corresponding to different AND-OR tree based models of concepts. That is to say, the same sub-goal has different sub-KBs in different stages of design.

Different design processes, different components (or parts) and different supporting tools for concurrent design have their own sub-knowledge bases (Sub-KBs). The describing model can be represented as following equation.



$$\text{DomainKnowledge} = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l \text{SubKB}_{ijk} \tag{1}$$

where, i is the serial number of processes; j is the serial number of components; and k is the serial number of supporting tools.

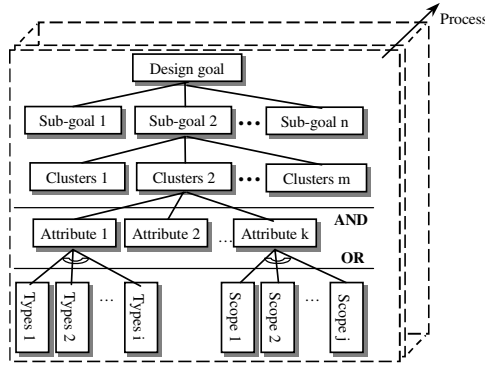


Fig. 4. The AND-OR tree based model of concept for the product design goal

#### 4.4 Integrated Knowledge Representation

The integration of knowledge representations is to combine all kinds of history knowledge above into a kind of unified KR schema, which includes the representations of process history, design history and domain knowledge. Generalized rule (GR) is used here whose premises and conclusions are extended to contain more kinds of representation forms of the history knowledge.

The formalization form for GR is shown as following:

- <GR> ::= <conclusion> ← <premise> <reliability>
- <premise > ::= <fact> | <burst condition>
- <conclusion> ::= <fact> | <process>
- <reliability> ::= <real which more than 0, less than or equal to 1>
- <process> ::= <method> | <model> | <modular of neural networks> | ...
- <modular of neural networks> ::= <input transactor> <neural networks> <output transactor>
- <method> | <model> ::= <function> | <process outside> | <dynamic link library>

Above all, an integrated history KR schema is presented with GR as shown in Figure 4. That is, with the control of meta-knowledge and the guidance of hierarchical framework, the domain knowledge is classified into a multi-level tree-structure according to the design process and design sub-goals, and its representation schema is GR which integrates rules, processes, methods, neural networks and so on into one. Thus a hierarchical KB is set up in which the framework is a multi-level frame and its sub-frames can inherit the knowledge and features from the father-frames.

## 5 Implementation

With the support of several projects from High Technology Research and Development Programme of China, a knowledge-based PDPM system is developed [6][17]. Then an IKM system is integrated into the PDPM system in the lifecycle of product development process and used in several enterprises.

### 5.1 Functional Modules of PDPM System

In the process-centered system, the integrated knowledge management system is attached to the PDPM system. Integrated to engineering design system, the knowledge system gets the design intent and history information of product with the help of PDM, and acquires the interactive information from designers and supporting tools in the environment of PDPM. It is the main part of PDPM to realize general IKM.

The knowledge-based PDPM system consists of role management, process analysis and improvement, process modeling, process executing, process monitoring, history management, coordination management, conflict management, knowledge acquisition, decision support and so on, which are integrated together on the workbench of the system. The functional modules of the PDPM system are shown in Figure 5.

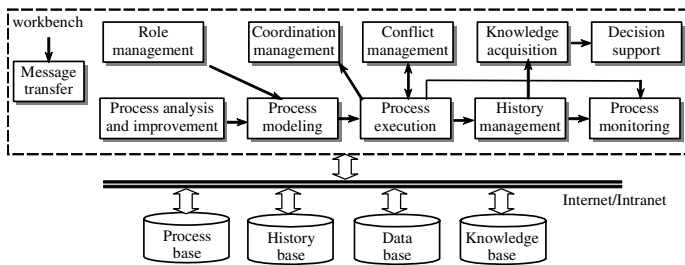


Fig. 5. The functional modules of knowledge-based PDPM system

In this system, the sub-systems of process execution and process monitoring drive the development process and update the process status. The sub-system of history captures and records the process history, design history, the status of using resources and so on. It can improve the project leader's insight to the product development and provide a foundation to improve the process later to record the actual process, decision and principle used.

### 5.2 Implementation of IKM System with Support of PDPM System

Figure 6 shows the implementation of IKM system with the support of PDPM system. It includes the input and query of design intent, input and query of domain knowledge and capture of process history. With the main clue of process history, the design history and the domain knowledge are acquired and input into knowledge base including process base, history base and knowledge base. The knowledge above is reused during the right sub-process at proper time by certain developer.

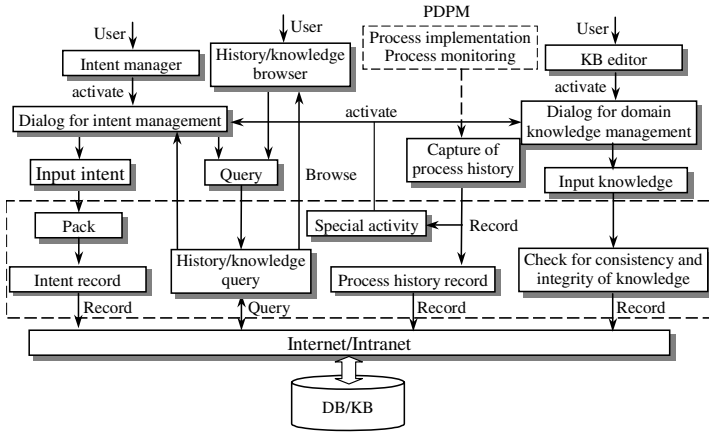


Fig. 6. The implementation of the IKM system

5.3 A Case Study

The integrated supporting environment of the IKM system based on PDDM is developed and has a Browser/Client/Server structure. It provides an ideal support environment for the history knowledge management and cooperative design with Internet/Intranet. The system has been used successfully during the life cycle of a new type of railway rolling stock development in QQHR railway rolling stock company and obtained satisfactory results.

Figure 7 shows one subsystem of process history and design intent for the development of grain railway rolling stock. The left part of the first dialogue box is a structural tree for the process to list all activities. The right part of the first dialogue box lists the history and intent for the related activities.

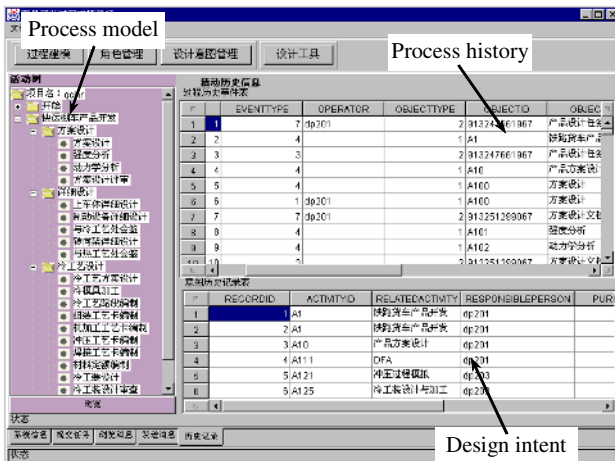


Fig. 7. The sub-system of process history and design intent

Supported by the IKM system, designers can browse and query the design history including process history, design intent and domain knowledge in the lifecycle of the product development process.

The product development process is compared with previous cases. It used to take about 6 months to develop a similar product before. With the help of knowledge-based PDPM system, it just took about 4 months to develop the product, that is, the time is shortened about one-third.

## 6 Conclusions

The integrated knowledge management in lifecycle of product development process is one of key enabled technologies to implement CE. Using the system correctly, the knowledge can be saved and reused, including the process history, design intent and domain knowledge in enterprises. It is an effective tool to support the knowledge-based PDPM system and intelligent design.

The further research work is to integrate the IKM system into PDM or add the function to PDM in order to realize managing knowledge more effectively and combine with CAD more closely. Another important aspect is to integrate the system and the knowledge-based engineering, that is, the knowledge can meet the requirement at the right time in suitable form during the lifecycle of the product development process. In conclusion, it is necessary to build a common framework to support the integration of concurrent PDPM and domain knowledge management in the lifecycle of the product, realize the close integration of development process, knowledge management and CAD to support the intelligent, networked, visual and numerical product design.

## Acknowledgments

The research is supported by National Natural Science Foundation of China - Methodology for domain knowledge management in cooperative product development (50275090) and The Scientific Research Encouragement Fond for Middle and Young Scientists of Shandong Province, China - Intelligent decision support system of complex product cooperative design for process (02BS071).

## References

1. Alho, K., Lassenius, C., Sulonen, R.: Process Enactment support in distributed environment. *Computers in Industry* 29 (1996) 5-13
2. Xue, D., Yadav, S., Norrie, D.H.: Knowledge Based and Database Representation for Intelligent Concurrent Design. *Computer-Aided Design* 31(1999) 131-145
3. Chen, Y.Z.: The Architecture of a Computer-Aided Collaborative Product Development Environment. *Proceedings of International Conference on Intelligent Manufacturing*, (1995) 302-324.
4. Zhong, P.S.: Knowledge-Based Process Management for Concurrent Product Development. Postdoctoral Research Report, Tsinghua University, (2001)

5. Wang, J.B.: Studies on Product Development Process in Concurrent Engineering. Ph.D. thesis, Tsinghua University, (2000)
6. Wang, J.B., Xiong, G.L., Chen, D.: Rule-Based Product Development Process Modeling with Concurrent Design Iterations Supported. *Journal of Tsinghua University (Science and Technology)*, 39 (1999) 114-117
7. Smith, R.P., Morrow, J.A.: Product development process modeling. *Design Studies*, 20 (1999) 237-261
8. Wu, Z.B., Xiong, G.L.: Research on a Management System of Concurrent Engineering. *High Technology Letters*, 6 (1996) 21-25
9. Maropoulos, P.G.: Review of research in tooling technology, process modeling and process planning, Part 1: Tooling and process modeling. *Computer Integrated Manufacturing Systems*, 8 (1995) 5-12
10. Chen, A., McGinnis, B., Ullman, D., Dieterich, T.: Design History Knowledge Representation and its Basic Computer Implementation. The 2nd International Conference on Design History and Methodology, ASME; 1990, 175-185.
11. Shah, J.J., Jeon, D.K., Urban, S.D., Bliznakov, P., Rogers, M.: Database Infrastructure for Supporting Engineering Design Histories. *Computer-Aided Design*, 28(5) (1996) 347-360
12. Goodwin, R. and Chung, P.W.H.: An Integrated Framework for representing Design History. *Applied Intelligence*, 7(2) (1997) 167-181
13. Kim, J.K.: Knowledge Acquisition for Knowledge-Based Decision Systems. *Applied Artificial Intelligence*, 11(2) (1997) 131-149
14. Taylor, W.A., Weimann, D.H., and Martin, P.J.: Knowledge Acquisition and Synthesis in a Multiple Source Multiple Domain Process Context. *Expert Systems with Applications*, 8 (1995) 295-302
15. Matratrinis, N.F., Doumpos, M., and Zopounidis, C.: Knowledge Acquisition and Representation for Expert Systems in the Field of Financial Analysis. *Expert Systems with Applications*, 12(2) (1997) 247-262
16. Murrell, S. and Plant, R.T.: A Survey of Tools for the Validation and Verification of Knowledge-Based Systems: 1985-1995. *Decision Support Systems*, 21 (1997) 307-323
17. Zhong P.S.: Knowledge Base System for Intelligent Decision Support of Concurrent Design. Ph.D. thesis, Harbin Institute of Technology, (1999)

# Knowledge-Based Cooperative Design Technology of Networked Manufacturing

Linfu Sun

CAD Engineering Center, Southwest Jiaotong University, Chengdu, 610031, China  
sunlf@vip.163.com

**Abstract.** According to the requirements of cooperative product development in networked manufacturing environments, a knowledge-based cooperative design technology is proposed. Based on the Chengdu-Deyang-Mianyang Networked Manufacturing and ASP Platform, a knowledge-based networked manufacturing cooperative design platform is developed by exploiting the product function modules, parts management, supplier management, customer management, and resources allocation optimization. In the cooperative design platform, the integration of manufacturing resources of supply chain and the method of client customization and enterprise cooperation are realized in the course of product development. The platform has been put into application in a number of enterprises in the Chengdu-Deyang-Mianyang region of China.

## 1 Introduction

With the advancement of society and the development of technologies, the requirements for innovative products have become higher and higher. People have changed from the pursuit of elementary products to the pursuit of complicated products that can produce economical, social and environmental benefits. Further more, they pursue artworks with cultural connotation that reflects ideology and social living style<sup>[1]</sup>. The innovation of the product development methods has been a powerful means that enterprises can achieve competitive advantages. First, the complexity of modern products makes it impossible for the products to be developed by an individual. Instead, it requires many people to cooperate. Second, the improvement of the product functions needs creative development methods and means that require the developers' knowledge and wisdom<sup>[1]</sup>. Third, the products and sociality of products development require people to organize and make use of resources all over the world, and quickly develop new products. Fourth, products and their economic and social benefits in the process of development make people take into account of not only the investment of product development, but also all the costs during the process of product operation and management as well as the costs and efficiency of the whole product lifecycle. Fifth, with the rapid development of information technologies and economic globalization, great changes have taken place in enterprise environments. Inter-industry cooperation across regions and countries, rapid development of united enterprises, and information technologies have resulted in a revolutionary reform of the management mode of product development methods.

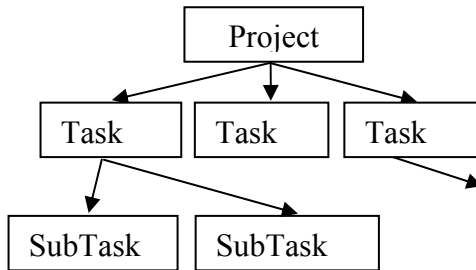
Cooperative design technology shows that the interaction, distribution, and coordination of people’s working mode have become one of the most active research areas of computer engineering applications. Supported by National High-tech Research & Development Program (863 Program), the research of cooperative design technology in China has advanced greatly. Beijing Qingruan information technology Ltd., Beijing Beihang Haier software Ltd., and other 8 Chinese companies have used their own 2D and 3D CAD systems in the development of a cooperation-supporting design system, integrating CAD, PDM/PLM, and other software systems to support cooperative product developments. A number of cooperative design-supporting creative technologies have achieved breakthroughs and several systems have been implemented, such as parts supplier management system technology in cooperative design<sup>[2]</sup>, mass customization products cooperative design system technology<sup>[3]</sup>, networked products cooperative design supporting system<sup>[4]</sup>, Web-based cooperative assemble system<sup>[5]</sup>, and complicated products cooperative manufacture supporting environment technology<sup>[6]</sup>.

## 2 Research on Enterprise Cooperative Mode

### 2.1 Traditional Cooperative Mode Based on Enterprise Internal Cooperation

The traditional design department and the organizing mode of design are usually hierarchical. The design room is directly leaded and administrated by the design department which is guided by enterprise managers. According to the attributes of the tasks, enterprises divide the project into sub-tasks and assign them to different design departments for execution, as showed in Fig. 1. This structure can be described easily by object-oriented method. This cooperative mode has such features as follows:

- (1) Enterprise can build uniform information system and cooperative architecture;
- (2) Fixed cooperative relation has been formed in enterprise;
- (3) Enterprise resources have been accumulated chronically, and design knowledge and experience in special domains have been formed systematically.
- (4) The cooperation pattern is relatively close, which makes enterprises lack information exchange with others.



**Fig. 1.** The task structure based on collaboration in an enterprise

## 2.2 Cooperation Patterns Based on Enterprises Cooperation

There are different kinds of cooperation patterns across the enterprises, including product design, product manufacture, product sale, product service, etc. Cooperative modes across enterprises include the cooperation pattern cross the enterprises, the cooperative mode of the leading enterprise regarded as the core, the cooperative mode on mass customization<sup>[3][7-10]</sup>.

## 2.3 Networked Manufacturing-Oriented Cooperative Mode

The cooperation based on networked manufacturing aims at the cooperation among regional convergent enterprise groups. Regional convergence includes scattered convergence, homogeneous convergence, and industrial-chain convergence. There are abundant resources in the region, but the systems and resources are heterogeneous and enterprises do not cooperate well with one another. The application of heterogeneous resources and the promotion of specialized cooperation is the key point of this cooperative mode.

# 3 The Design of Cooperative Design Platform

## 3.1 Functions of Cooperative Design Platform

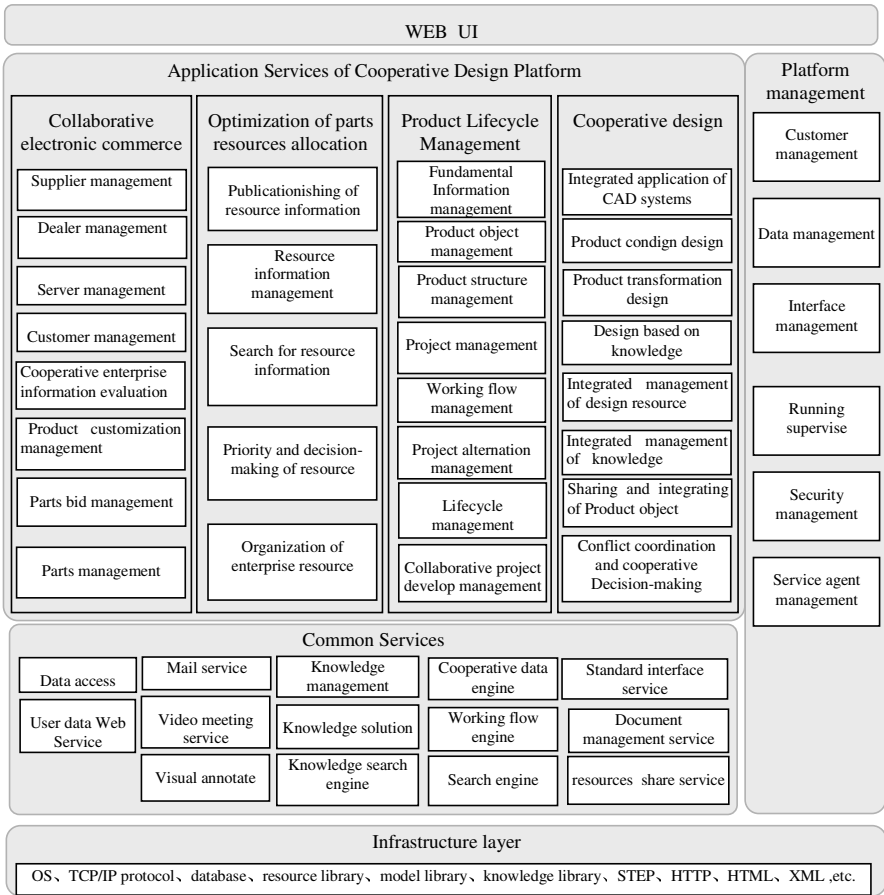
Cooperative design mode lies on the enterprise cooperative mode. According to the study of enterprise cooperative mode, cooperative design systems mainly focus on the following aspects:

- (1) Distributed synchronized cooperation<sup>[9][10]</sup>.
- (2) Design based on knowledge<sup>[11]</sup>. First, the cooperative design system is a system which supports cooperative “design”. Then, it is a system which supports “cooperation”. The cooperative design requires the application knowledge and experience, so the use of the application knowledge and the support for the process of product design are the key issues of the cooperative design systems.
- (3) The integration of resources within a supply chain. Modern enterprises need implementation of specialized work and cooperation, and integration of all the resources within the supply chain, so it can respond rapidly to the changes of market demands.
- (4) Participation and customization of customers<sup>[3]</sup>. Under the customized manufacturing mode, the cooperative design system should provide an efficient communication platform for customers, designers, and companies, implement real-time communications, and respond to the customer requirements quickly.
- (5) Product lifecycle management. In the process of product design, mass data are supervised by the PDM system and many enterprises will be involved in the product development. Because the relation between corporations is equal, the enterprise organization’s hierarchy in hypo-taxis relationship cannot be formed. Thus, it is essential to support product lifecycle management.



### 3.2 The Architecture of the Cooperative Design Platform

Fig. 2 shows the proposed architecture of the cooperative design platform of networked manufacturing. The platform uses open system framework, and is based on the design of layered system architecture.



**Fig. 2.** The cooperative design platform architecture of Networked Manufacturing

- (1) Infrastructure layer: the platform’s infrastructure utilizes TCP/IP services, shields various physical connection attributes, and provides transparent information communication service. This layer includes database, resource library, model library, and knowledge base of the heterogeneous distributed information systems, integrates various CAD systems, parts resource and design knowledge in enterprises, and also provides domain service, directory service, component object service, message service, and other services for the platform. The platform uses standard system software. The diversity of supports to the system software shows the compatibility of the platform.

- (2) Common service layer: common service layer comprises a set of common kernel components. Functions of these components are limited to the internal use of cooperative design or collaborative electronic commerce. They provide general infrastructure services for the whole cooperative design platform.
- (3) Application Service layer: application service layer is the main body of the cooperative design platform. It comprises four modules: cooperative design, product lifecycle management, optimization of parts resources allocation, and collaborative electronic commerce. Cooperative design module is the core of the whole platform, which comprises the allocation design of products that customers participate in, the transferring product design based on existed parts, and the creative product design based on knowledge. It can use various CAD systems' resources libraries and provide general tools for cooperative design. It can use various methods to realize the distributed synchronized cooperation, the mechanism of conflict coordination and cooperative decision-making, the management of product lifecycle data, and the complex management of cooperative design and data of subsequent process of manufacturing, sale and service. It supports integration of the platform framework, object management service of distributed transactions, and application integration in an enterprise or across multiple enterprises. Optimization of parts resources allocation module searches the parts information for products in resource library, optimizes and configures the parts. Collaborative electronic commerce module provides tools for the integration of the resources of entire enterprise supply chain and resources of regional enterprise group in the process of cooperative design.
- (4) Web UI: to provide uniform and secure user interfaces for users. They enable users from different locations and different identities to access various services of the cooperative platform through the same interfaces.

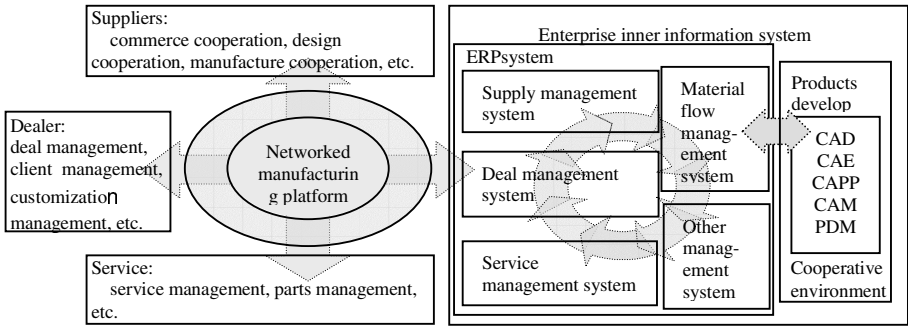
## **4 Networked Manufacturing-Oriented Cooperative Design**

### **4.1 Establishment of Collaborative Relationship of Manufacturing Industrial-Chain**

Collaborative relationship is centered on the leading enterprise and founded by the leading enterprise and its united enterprises by dynamic alliance, as showed in Fig. 3. United enterprises mainly include parts suppliers, products dealers, and services. Networked manufacturing may have a group of leading enterprises so as to support the collaborative system of socialization of manufacturing industrial-chain.

Cooperative design platform can provide product designers with "ideal" suppliers and "ideal" parts through the suppliers management, and evaluate suppliers and parts through networked invitation of public bidding which enrich design resources enormously. Following the rules of commerce and technology, designers can choose the most appropriate parts. Buyers can manage parts and suppliers conveniently and consolidate the relationship with suppliers.

Gao et al. [2] presented a choosing and evaluating system of the suppliers based on enterprise competition. Evaluation criteria include: competition power of products, competition power within the enterprise, competition power outside the enterprise, ability of cooperation.



**Fig. 3.** Manufacturing industrial-chain model based on enterprise

The cooperative relationship management of manufacturing industrial-chain is performed by the cooperative commerce module which includes: parts inventory management, product sales management, service management, etc., and has such functions as follows: inventory management, inventory contract management, sales management, social storage management, client repository management, product repository management, maintenance records management, and parts/used-parts management, etc., which become the foundation of the cooperative design platform development.

**4.2 Optimized Configuration of the Enterprise Parts Resources**

The priority of resources is an important aspect of cooperative product development. On the platform of regional networked manufacturing, we developed resource configuration center and resource optimized configuration system. The module was developed by Huazhong Technology University and Southwest Jiaotong University. The resource optimized configuration system has been developed with the hierarchy analysis method. We can make optimization and evaluation through five criteria: machining time, machining cost, reliability, precision, and credit of enterprises. When a client selects the resource on the platform, he first compares the relative importance of the five optimized goals, then gets their relative coefficient, finally gets the synthesis score. This is an effective method to solve the optimization problem of big system with multiple hierarchies and multiple goals.

**4.3 Customization of Products**

The clients' network customization can be implemented by the platform through issuing configuration information of products and providing products configuration model online. Enterprises check the finished products storeroom, work out production plan, and organize the production process through the customized order form. The platform supports the customized management of products through integrating the inner information system within the enterprise: taking into account the user's request and production plan; supervising the process of design, manufacturing, in/out storeroom of products, and referring users; so as to trace the production process of

customized products and monitor the production status of customized order form in real time.

#### **4.4 Design and Manufacture Collaboration of Enterprises**

The platform has such functions as follows: cooperative design between enterprises and suppliers, project management, document management, drawings approval management, collision management, and other cooperative tools. Among them, project management module manages design process and classification of products, parts, standard parts, and establishes examining flow and product relationship model. Document management module is used for drawings browse, document query, version management, document referring, and downloading. Drawings approval module can provide effective examination flow control, drawings browse, annotation and associated model query. Collision management module manages every part's interface dimension to insure its assemblability.

#### **4.5 Product Lifecycle Management**

The platform is integrated with the internal information system of the enterprise, and makes use of the returned information in processes of product design and manufacturing, parts inventory management, sales management, and after-sale-services. These help establish client repository and product repository, maintain the related databases, and realize the total management of databases of products and parts. Through the platform and its service management system, a manufacturing enterprise can find services providers and parts suppliers easily; and parts suppliers can also monitor their parts and reclaim the used parts to manage the used ones efficiently. So the platform has been the hinge of enterprises and parts suppliers and services where the business information can be exchanged, thus we can support the lifecycle of products and parts.

### **5 Design Technology Based on Knowledge<sup>[11][12]</sup>**

A cooperative design system is first a system which supports cooperative "design". Due to a long time commercial consideration, CAD technology emphasizes on the system's all-purpose, but ignores character of the designed object, design process, and the Web technology functions on increasing the design level and efficiency. Thus, traditional CAD systems mainly focused on modeling of parts, lacking effective supports for product design. It makes the input data massive, and operation steps complex. Furthermore, it cannot directly relate the parts model with the assembly model. The model becomes very complex and is difficult to maintain. It cannot effectively solve design problems about layout and connection and coordination of parts during conceptual design phases.

Product design is a complex process which is summarized by knowledge accumulated for many years. We can see that: (1) Practical product design process is not a design process which begins with lower level geometry model. Designer has some concept in the earlier stage of design. The concept is the consequence of special field thought pattern, which guides the whole design process. (2) In the process of

design, the product structure design parameters which the experts concern for having their special meanings. These parameters are the results developed in special fields after a long time. Thus they always cannot be expressed clearly by geometry model. (3) Product design is itself a morbid construction. It cannot be modeled by full formulas. A design process is not a pure logical process. The design knowledge and experience are also very important.

### 5.1 The Structure of Product Object Model

Attributes of a product are themselves infinite, but the people's knowledge about them is finite. Design process oriented system needs design parameters that experts are concerned. If  $S$  denotes a certain state of the structure,  $X$  shows its attribute, as follows:

$$S_i = ( X_1(t_i), X_2(t_i), \dots, X_n(t_i) ) \quad (1)$$

Where  $t$  denotes time,  $X_j(t_i)$  is the state of  $X_j$  at some time,  $\Gamma_i$  shows an operation exert on state  $S_i$  which brings about a state transfer, and  $\Sigma$  shows the aggregate of all operations.

$$S_{i+1} = \Gamma_i S_i \quad (2)$$

$$\Sigma = \{ \Gamma_1, \Gamma_2, \dots, \Gamma_n \} \quad (3)$$

Meantime, whether engineering structure or operations exerted on state will be restricted, there are constrains on attributes and operations. If we use  $\Pi$  to denote the operations which satisfies the algorithm,  $V_i$  shows value field when  $X_i$  is restricted by a variable, then state model of engineering structure is defined as:

$$D_s = \{ S_s, S_o(X_i) \mid X_i \in V_i, \Sigma(\Gamma_i) \mid \Gamma_i \in \Pi \} \quad (4)$$

Where  $S_s$  shows preliminary state of the design,  $S_o$  shows objective state of the design,  $D_f$  shows function description with the design object,  $K_w$  shows knowledge exerts on object, and  $R_p$  shows formation pattern of design object. Then, design object  $O_j$  is represented as follows:

$$O_j = \langle D_f, D_s, K_w, R_p \rangle \quad (5)$$

### 5.2 Engineering Knowledge and Expression

Engineering design knowledge is closely related to specialized domains, and the characteristic of engineering thought modes and engineering data should be taken into account. There are many types of engineering knowledge. From logic aspect, it can be divided into design object properties and relations, object development rules and design controlled-process knowledge, skills or experience knowledge, design common sense, and design knowledge management strategies, etc. From the aspect of knowledge property, it can be divided into static knowledge, which describes design object, and dynamic knowledge of design process. From the aspect of gaining access, it can be divided into example knowledge, engineering criterion knowledge, design

experience knowledge, etc. The author has developed an engineering design knowledge representation system described in [12].

### 5.3 Product Design Process Modeling

Product design problems are complex. When generation mode system is used to solve these problems, the designs usually cannot be completed because of the problems' overall scale. Therefore, the author proposed using the Programming-controlled Two-stage Design Theory to complete the CAD system's development. The programming is a kind of decomposed technology, and through decomposition, products design forms hierarchical structure which consists of parent classes and child classes. The design process of the second stage is a target searching process. Thanks to the first stage design, it is considered that the searching process is completed in the most possible solution-obtained state, making the searching process easier.

*(1) Initial solution of the general design.*

For the general design, the first process is shown as in formula (6).

$$D = \Gamma ( D_s, D_f ) \tag{6}$$

$\Gamma$  is a transform operation and  $D$  is the initial solution.

*(2) Initial solution of the creative design*

Pre-defined object cannot meet the design requirement directly in the creative product design, a part of it has surpassed the design space defined by the object model, which needs improvement and expansion. Creative design needs model matching. Further improvement and expansion are needed in object class knowledge, design method, application goal, etc., which enable the creative design to meet the requirements of the new problems.

$$K'_w = \Gamma (K_w) \tag{7}$$

$$D'_f = \Gamma (D_f) \tag{8}$$

$$R'_p = \Gamma (R_p) \tag{9}$$

$$D'_s = \Gamma (K'_w, D'_f, R'_p, D_s) \tag{10}$$

$$D = \Gamma (D'_s, D'_f) \tag{11}$$

### 5.4 Cooperative Design Based on Knowledge

Based on the product object model and design process model discussed above, the knowledge-based cooperative design system must be supported by “product database”, “design model database”, and “design knowledge base”. The three databases are composed of design objects which can be used to describe products and support designs. Combined with these, based on the above formulas from (1) to (6), knowledge management system, knowledge solution system, and knowledge search

engine are developed to finish the knowledge-based cooperative design. Then we can achieve the design of knowledge-based cooperative system. The following aspects are considered:

- (1) Deduction based on project instance-class knowledge: After analyzing the entire instance model, we can deduce the particularly designed parameters with some characters we know. This method is used to build multi-layer evaluation model with fuzzy matching model and fuzzy evaluation model, and we can match the programs of the whole project with the relation programs of instance in knowledge storage.
- (2) Deduction based on project-language class knowledge: we can build integrity project-language knowledge model and then establish project-language knowledge, and with these develop project-language deduction engine. Thus we can express and deduce all kinds of project knowledge and expert's experience-knowledge.
- (3) Deduction based on project data-table class knowledge: with the building of known area, conclusion area, and notation area, we can deduce the related design parameters by using the known conditions and the attributes of the data-table.

## 6 Conclusion

Based on the discussions above, Southwest Jiaotong University and the Advanced Productivity Center of Manufacturing Information of Sichuan have built a cooperative design platform, and achieved the integration with the Chengdu-Deyang-Mianyang Networked Manufacturing and ASP Platform. Chengdu-Deyang-Mianyang Networked Manufacturing and ASP Platform includes the cooperation center of manufacturing product chain, resource-configuration center, technology-support center, information center, and management center. Here resource-configuration center achieves the classification management of the resources, the integration and sharing of the software resources, the optimized and technical resource integration of equipment resources; the cooperation center of manufacturing product chain achieves the cooperative relation of the main manufacture enterprise and their suppliers, dealers, and service stations.

The cooperative design platform has become the important part of the center of manufacturing cooperative chains. With this platform, we have achieved the cooperative system in some motor manufacturing, for example, WANGPAI Motor Inc. of Chengdu and YUNNEI Engine Dazhou Motor Inc. The management of manufacturing and parts suppliers, product dealers, service stations have also been built.

The clients' network customization can be implemented by the platform through issuing configuration information of products and providing products configuration models online. The platform supports the customized management of products through its integrating inner information system with the enterprise: taking into account the user's requests and production plan; supervising the process of design, manufacturing, in/out storerooms of products and referring users. Fig. 4. shows the customized motor model of Chengdu WANGPAI Motor Inc.



Fig. 4. Shows motor customized model of Chengdu WANGPAI motor Int

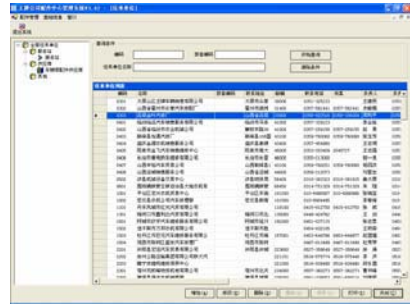


Fig. 5. Management of the parts center

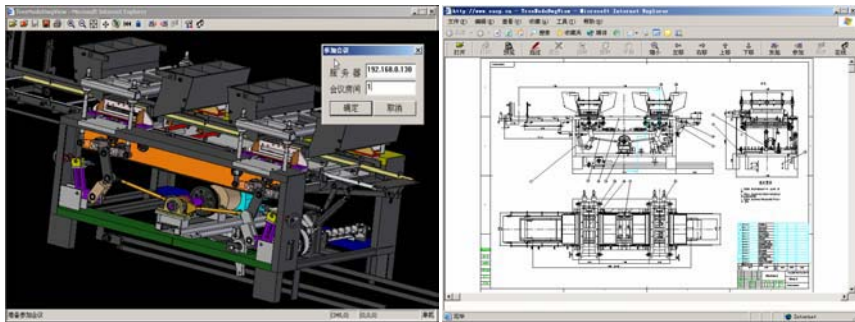


Fig. 6. Production line of assembly chart based on platform

We have built customer repository and product repository on this platform. Using these repositories, we achieved the matchmaking of the customers and suppliers of parts and services. During the service program, the suppliers can reclaim used parts. Fig. 5 is the management of the parts center.

We have also designed the program management, the document management, the checkup graphic management, and the conflict management, and thus we can sort products, parts, and any other standard components. We can also use this platform to browse graphic models, check out documents, manage systems, control processes of testing and approving, and check out document annotation and relation model. Fig. 6 shows the production line of assembly chart based on this platform.

### Acknowledgement

This paper is supported by National High-Tech Research and Development Program (863 Program) of China.



## References

1. Sun, L.: Digital Design Technology for Product-developing Process, *Computer Integrated Manufacturing Systems-CIMS*, 9(12) (2003) 1088-1091(in Chinese)
2. Gao, L., Tong, B., Dong, X.: Research on Component and Supplier Management in Cooperative Design, *Computer Integrated Manufacturing Systems-CIMS*, 8(10) (2002) 766-769 (in Chinese)
3. Chen, C., Zhang, S., Zhang, J., Wang, B., Li, L.: Product Cooperative Design System Based on Mass Customization, *Computer Integrated Manufacturing Systems-CIMS*, 9(9) (2003) 788-792 (in Chinese)
4. Tian, L. and Tong, B.: Design and Development of Net-based Cooperative Product Design Support System, *Computer Integrated Manufacturing Systems-CIMS*, 9(12) (2003) 1097-1104 (in Chinese)
5. Dong, X., Tong, B., Gao, L., Luo, W.: Research on Web-based Cooperative Assembly System and Its Related Key Technologies, *Computer Integrated Manufacturing Systems-CIMS*, 9(1) (2003) 20-24 (in Chinese)
6. Li, B., Chai, X., Zhu, W., Suen, J., Liang, B., Wu, H., Peng, X.: Supporting Environment Technology for Cooperative Manufacturing of Complex Product, *Computer Integrated Manufacturing Systems-CIMS*, 9(8) (2003) 691-697 (in Chinese)
7. Shyamsundar, N., Gadh, R.: Internet-based collaborative product design with assembly features and virtual design spaces, *Computer-Aided Design*, 33(9)(2001) 637-651
8. Regli, W.: Internet-enabled computer-aided design, *IEEE Internet Computing*, 1(1) (1997) 39-51
9. Maher, M.L., Rutherford, J.H.: A Model for Synchronous Collaborative Design using CAD and Database Management, *Research in Engineering Design*, 9(7) (1997) 95-98
10. Pahng, F., Senin, N., Wallace, D.R.: Distributed modeling and evaluation of product design problems, *Computer-Aided Design*, 30(5) (1998) 411-423
11. Sun, L.: The Development of Knowledge-Based Intelligent CAD Systems, *Journal of Southwest Jiaotong University*, 34(6) (1999) 611-616 (in Chinese)
12. Sun, L.: The Knowledge Representation System for Engineering Design, *Journal of Southwest Jiaotong University*, 34(6) (1999) 617-624 (in Chinese)

# Multi-ontology Based System for Distributed Configuration

Xiangjun Fu<sup>1</sup> and Shanping Li<sup>2</sup>

<sup>1</sup> School of Mechanical Engineering, Shanghai Jiaotong University,  
200030, Shanghai, China

<sup>2</sup> Institute of Artificial Intelligence, Zhejiang University,  
310027, Hangzhou, China

fuxiangjun@hotmail.com, shan@cs.zju.edu.cn

**Abstract.** Online configuration for products in the distributed and dynamic computing environment motivates the demand for semantic based cooperation, which takes place in the supply chain under B2B situation. Traditional stand-alone knowledge model of configuration systems does not meet the new requirements. We propose a multi-ontology based solution. The core idea of this paper is to take the process knowledge of distributed systems into consideration, which supports the integration among distributed configuration systems. Furthermore, the process model provides possibility to optimize solutions for configuration problems. OWL is used as modeling language in order to utilize potential benefits of current Semantic Web technology.

## 1 Introduction

Configuration service calculates product variants, which fulfill customer requirements as well as technical and non-technical constraints on the product solution. Due to highly specialized economy and rush for supply chain integration by Web-based procurement, joint configuration by multiple business partners is becoming a key enabler of the mass customization paradigm [1]. Many research and commercial configuration systems have been developed. Previous developed systems, like R1/XCON [2][3], are rule based. This approach requires the rule database to grow with products involvement, and can lead to huge, barely manageable systems. For this reason, rule-based configuration is currently losing ground to model-based systems. The main assumption behind model-based systems is the existence of a system's model, which consists of decomposable entities and interactions between their elements. Model-based systems possess the advantages of enhanced robustness, enhanced compositionality and enhanced reusability [4]. There are several model-based approaches to configuration, such as logic-based approach, resource-based approach and constraint-based approach. Typically, the stand-alone knowledge models of these systems do not meet the new requirements imposed by online configuration in a Web of cooperating product and service providers. Several configuration tools [5][6] had been deployed on the Web. However the major obstacles to incorporating configuration technology in eCommerce environments are not addressed:

- Distributed and dynamic computing environments: Due to the frequent reconstructing of the network infrastructure, appearing and disappearing of suppliers, and cooperating privacy reasons, a simple solution with one single centralized knowledge base and problem solver is impossible.
- Heterogeneity of configuration knowledge mode: Local configuration systems use different knowledge representation languages to model the configurations knowledge in syntax, use different ontologies to express the semantically same concept in the model, and employ some specialized solutions for their task's characteristic.

The emerging semantic web technology, which aims at improving the “semantic awareness” of computers connected via the Internet, provides a platform for the distributed configuration systems of multiple suppliers to collaborate on products and services. Our goal is to utilize semantic web technology to facilitate the development of configuration in the aspects mentioned above, which will be crucial to the success of distributed configuration systems. A framework is proposed to build the configuration system on the semantic web [7]. Compared with the framework proposed in [7] and different from existing configuration models [8][9], this paper takes the process knowledge into consideration. Process model is essential for distributed configurations cooperation. Furthermore it also provides possibility to optimize solutions for configuration questions. Multileveled ontology approach is employed to model the knowledge of the varied configuration services. Multileveled ontology divides the knowledge of configuration system into general level and domain specified level, as shown in Figure 1. Through the general level ontology, local systems even with heterogeneous models can realize information sharing and integration.

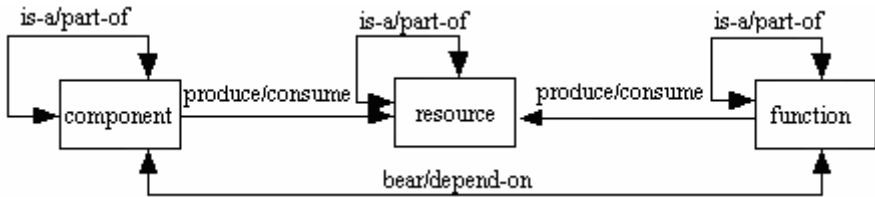


Fig. 1. The general ontology

## 2 The Generic Ontology Level

An ontology is an explicit specification of a conceptualization [10]. Therefore it is the couching of knowledge about the world in terms of entities (things, the relationships and constraints holding among them). Typically, an ontology contains hierarchical descriptions of important concepts in a domain, and describes crucial properties of each concept through an attribute-value mechanism. Additionally, further relations among concepts may be described through additional logical sentences. Finally, individuals in the domain of interest are assigned to one or more concepts in order to give them proper type. Through defining the shared and common domain theories,

ontologies help to communicate concisely—supporting semantics exchange, not just syntax exchange. Built on top of them, a higher-level ontology (with complex logics and the exchange of proofs to establish trust relations) will enable even more powerful function. In order to define a unified configuration model, we extract a set of generic concepts from the configuration models defined in [8][9] and build a generic ontology which provides the formal vocabulary to specify the configuration model of the next level: application domain oriented configuration ontology. Concepts in the generic ontology can be divided into three parts:

- Conceptual Knowledge Objects in the application domain are described by means of concepts. Concepts are represented through a name and their properties (i.e., parameters and relations to other concepts). Taxonomic structuring is achieved by specializations. Compositional structuring is described by aggregation (i.e., part-of). Restrictions between multiple concepts and their properties are expressed by means of constraints or rules.
- Procedural knowledge about the configuration process describes the ordering and execution of configuration decisions, the focus on particular concepts and conflict resolution methods. In this paper, process ontology is described in detail.
- Task specification describes the configuration goal. This specifies the demands a created configuration instance has to accomplish. The goal, which is put forwarded by client, may be functional requirement or structure of product. But the configuration instance is always described by the components, which construct the desired product. So the mapping from product structure to the function is needed. And in the generic ontology we must take concept of function and the mapping into consideration.

Following concepts are the basic parts of the generic ontology:

- Components. They represent parts the final product can be built of. Components are characterized by attributes that have a predefined domain of possible value. We classify components into complex components, atomic components and concrete components. Atomic components are the basic building blocks of configurations; complex components are structured entities whose characterization is given in terms of subparts which can be complex components in their turn or atomic ones; while concrete components refer to the entities whose attribute values are all from concrete domain, like number and string, which are the abstract descriptions of data structure.
- Functions. They are used to model the functional structure of an artifact. Similar to components, they can be characterized by attributes and can also be classified into complex functions and atomic functions. There exist bear and dependency relationships between components with functions.
- Resources. Parts of configuration problems can be seen as resource-balancing tasks, where some of the components or functions produce resources and others are consumers. Units serve to measure quantities of the same resource type. For clearly defining the amount of resources and calculation, units ontology also is a subpart for the general configuration. There exist consume and produce relationships between components/functions with resources.

- Generalization. Components/functions/resources with similar structures are arranged in a generalization hierarchy, i.e., is-a relationship between component/function/resource. And, generalization relationships are not hold among resources, functions and components.
- Aggregation. Aggregations between components/functions represented by part-of relationships describe a range that how many subparts an aggregation can consist of. But it needs to note that the aggregation relationship holds only when the associated components/functions can be composed to a new one, which owns different attributes from its members. For example, the part-of relationship holds between a computer and its sub components because a computer has the processing power that is only the whole computer holds.
- Ports. In addition to the amount of different components, the product topology may be of interest in a final configuration, i.e., how the components are interconnected with each other. In the definition of component, the attribute which refers to other component of the product is connection or port, which describes the relationships between components. Connections and ports are the important attributes between components. To describe whether the connections/ports attributes must be filled by other components in the configuration, we use the meta attribute optional/compulsive to express this knowledge.
- Constraints. In a complex product, some types of components cannot be used together or must concurrently appear in the configuration instance. Because this relationship cannot be elegantly described by connections/ports, we use sets of constraints to specify it.

So far, we have defined the basic concepts to describe knowledge of products and tasks, which reflects the static state aspect in the configuration model. Next we will discuss the dynamic part: the configuration process itself.

### 3 The Configuration Process Ontology

To achieve the integration between distributed configuration systems, two problems need to be solved: (1) how to efficiently discover co-configuration agents based on functional and operational requirements; (2) how to facilitate the interoperability of heterogeneous cooperators. Within the context of the emerging semantic web and the developing of workflow technology, we argue that configuration process ontology is needed. Specified based on Semantic Web Service ontology [11], the process ontology defines the common used concepts for configuration process, as show in Figure 2.

#### 3.1 Distributed Configuration Process Analysis

An execution of a configuration task can be considered as a workflow, where the valid requirements specification (RS) is the input and a satisfied configuration instance (CONFI) is the result worked out through the problem solving subsystems. So a local configuration problem (index by “i”) can be described by a triple  $(DD_i, RS_i, CONFI_i)$ , where DD represents the domain description, which includes the multilevel

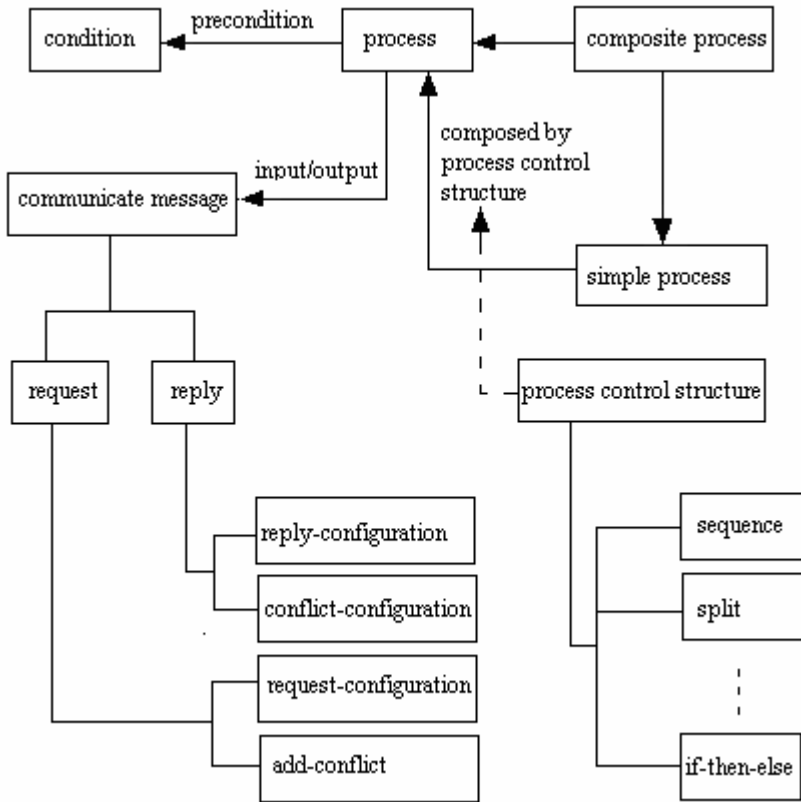


Fig. 2. Process ontology

Table 1. Process control structure

Process Control Structure	Execute a list of process in a sequential order
Sequence	Execute elements of a bag of processes concurrently
Split	Invoke elements of a bag of processes
Join-Split	Invoke elements of a bag of processes and synchronize
If-Then-Else	If specified condition holds, execute "Then", else execute "Else"
Unordered	Execute all processes in a bag in any order
Choice	Choose between alternatives and execute
Repeat-Until	Iterate execution of a bag of processes until a condition holds
Repeat-While	Iterate execution of a bag of processes while a condition holds

configuration ontology; RS represents the requirements specified by clients or other cooperators; CONF<sub>I</sub> is described by the assertions whose predicates symbols are in the concept set in application domain ontology. According to [12], in the general cooperation procedure of distributed configuration systems the communication messages have the following signatures: Request<sub>k</sub><sup>no</sup>(CONF<sub>I</sub><sup>si</sup>), the configurator “k” receives the configuration CONF<sub>I</sub><sup>si</sup> and checks if it is locally satisfiable. “si” denotes the search depth of the general procedure and “no” counts the interaction cycles.

- Reply<sub>k</sub><sup>no</sup>(CONF<sub>I</sub><sup>si+1</sup>). Configurator “k” communicates the configuration result CONF<sub>I</sub><sup>si+1</sup> in reply to Request<sub>k</sub><sup>no</sup>(CONF<sub>I</sub><sup>si</sup>) back to the facilitator. CONF<sub>I</sub><sup>si+1</sup> is a valid local configuration of configurator “k”.
- Conflict<sub>k</sub>(CONF<sub>I</sub><sup>sj</sup>). With this message the configurator “k” alerts the facilitator that CONF<sub>I</sub><sup>sj</sup> is not satisfiable with its local knowledge base.
- Add-conflict(C). Once the facilitator is alerted with a conflict message, it broadcasts this conflict C to all configuration agents. That is then negated and added to their local system requirements RS.

First, the configuration instance assertion sets are initialized and the facilitator agent distributes the non-empty sets of RS to the configuration agents. Then the facilitator starts the problem solving process by broadcasting Request<sub>k</sub><sup>1</sup>(CONF<sub>I</sub><sup>0</sup>) to each recipient of a non-empty RS<sub>k</sub> and awaits Reply<sub>k</sub><sup>1</sup>(CONF<sub>I</sub><sup>1</sup>) or Conflict<sub>k</sub><sup>1</sup>(CONF<sub>I</sub><sup>1</sup>) messages from these cooperating agents. If there exists a local agent replying with a Conflict<sub>k</sub>(CONF<sub>I</sub><sup>sj</sup>) message the facilitator will choose a conflict resolution strategy which needs to send message Add-conflict(CONF<sub>I</sub><sup>sj</sup>) to local agents and backtracks by demand another reply to Request<sub>k</sub><sup>no</sup>(CONF<sub>I</sub><sup>sj-1</sup>); Else, after collection of replies the facilitator unifies the locally completed results and broadcasts them to local configurators with a Request<sub>k</sub><sup>1</sup>(U<sub>k</sub>CONF<sub>I</sub><sup>si</sup>) message. The procedure repeats until terminated either with a valid solution or the systems detects that there exists no solution. This is only a rough process for the configuration task. To realize a practicable system, we still need both the ontologies of process structure and process control structure.

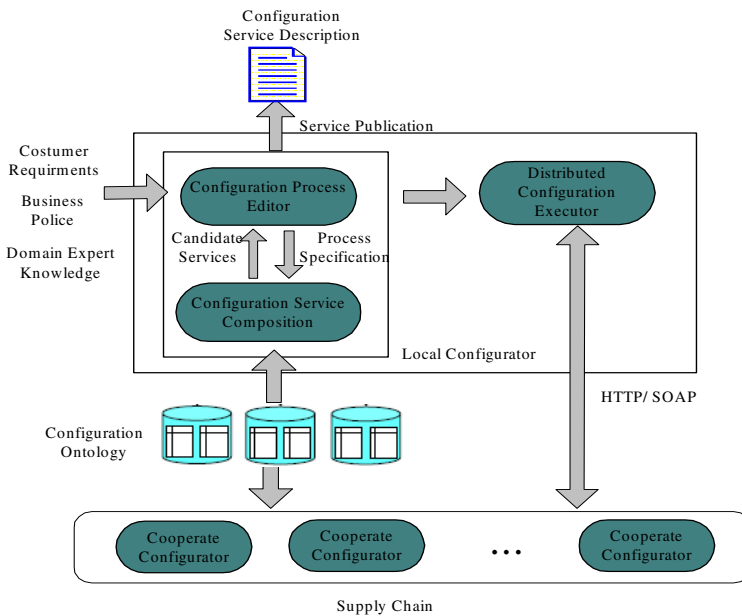
### 3.2 Configuration Process Ontology

The process structure ontology describes a configuration system in terms of its inputs, outputs, preconditions, effects and its component sub processes. Process control ontology describes each process in terms of its states, including initial activation, execution, and completion. We expect the configuration process ontology to serve as the basis for combining a wide array of distributed configuration services, and process control ontology to serve as the basis for the facilitator to use other workflow technologies, such as Petri-net, to get plan the problem solving task optimized and more effective. In developing the ontology, we drew from a variety of sources, especially in the emerging standards in process modeling and workflow technology such as the NIST’s Process Specification Language (PSL) [13]. The primary kind of entity in the configuration process ontology is unsurprisingly a process. A process can have input message representing the requirement or notice submitted by cooperators or clients, output message representing the answer message in responding to input message. Besides inputs and outputs, there can also be precondition, which must be held for the process to be invoked. To describe the modularization of configuration process, we

distinguish between atomic and composite configuration processes. Atomic processes are directly inviolable, have no sub processes, and are executed in a single step from the perspective of the facilitator. Composite processes are decomposable into other processes or simple processes. Their decompositions are specified by using control constructs such as sequence and if-then-else. A composite process must have a composite property by which the control structure of the composite is indicated. Each control construct, in turn, is associated with an additional property to indicate the ordering and conditional execution of the sub processes of which it is composed. For instance, the control construct, sequence is to associate the main process with a list of sub-processes which are to be executed orderly.

## 4 The Architecture of Configuration System

As shown in Figure 3, the architecture includes three modules: configuration process editor, configuration service composition, and distributed configuration executor. Cooperative configuration activities often involve constructing a workflow which



**Fig. 3.** The architecture of distributed configuration system

assigns subtasks among the whole supply chain. The workflow design is presided over by the sponsor of the supply chain. According to its own business policies, client's requirement and domain expert's knowledge, and guided by the process editor's graphical interfaces, sponsor designs the workflow for the configuration activities. The result of the design is a detailed process specification, which grounds on the basic concepts from general configuration ontology and domain ontology. As the instance



of up level configuration process ontology, it is described by OWL [14] document. Based on XML and RDF syntax, OWL is widely adopted as Semantic Web modeling language for ontology development.

```

<owl:ontology rdf:ID="PC_CONFIGURATION">
...
<owl:Class rdf:ID="PC">
  <rdfs:subClassOf rdf:resource="#ComponentType"/>
</owl:Class>
<owl:Class rdf:ID="hd-capacity">
  <rdfs:subClassOf rdf:resource="#Resource">
  <owl:onProperty rdf:resource="#volume"/>
</owl:Class>
<owl:Class rdf:ID="hd-unit">
  <rdfs:subClassOf rdf:resource="#ComponentType">
  <CONFIGURATION:parts_of rdf:resource="#PC"/>
  <owl:onProperty rdf:resource="#produce"/>
</owl:Class>
<owl:Class rdf:ID="motherboard">
  <rdfs:subClassOf rdf:resource="#ComponentType"/>
  <CONFIGURATION:parts_of rdf:resource="#PC"/>
</owl:Class>
...
<owl:Class rdf:ID="OS">
  <rdfs:subClassOf rdf:resource="#ComponentType"/>
  <CONFIGURATION:parts_of rdf:resource="#PC"/>
  <owl:onProperty rdf:resource="#require"/>
  <owl:onProperty rdf:resource="#consume"/>
</owl:Class>
...

```

**Fig. 4.** Computer components ontology

Taking process specification as inputs, the configuration service composition searches for suitable services, which are published by candidate cooperators. Services composition is based on the semantic matching among process's inputs and outputs. The logic base of OWL is description logics (DLs). Therefore the DLs reasoning engine—Racer [15] is used to retrieve sub-configuration services automatically that match the semantic of sub-process specification in the OWL documents. External agents can use the outcome of reasoning engines to select a configuration service with respect to the whole configuration task. But for some complex products or services, where all the knowledge needed for configuration cannot be captured explicitly, ontology-driven reasoning proves inadequate. So service composition module provides the interface for the expert to intervene the service composition procedure.

After all the cooperators and their services in the supply chain are decided, the distributed configuration executor invokes all the sub-configuration services

synchronously. And the distributed configuration algorithm is running at the sponsor site. Mediation messages are transmitted through the HTTP protocol.

## 5 Application Example

For comparing with the work which other researchers have done, we also introduce our framework of distributed configuration systems by presenting a motivating scenario from the customized personal computer (PC) selling. For a computer, besides its basic functions, a set of extended customized functions can also be obtained on the customer's will. The customer can choose among various constraints. Different subsidiary companies or the third parties usually provide these services and their parts, while the customer wishes to order a completely configured PC solution. The sponsor (local configurator) possesses strong mediation service enabling distributed configuration services to cooperate.

```

...
<process:AtomicProcess rdf:ID="pc-parts-configurationA">
  <process:hasInput>
    <process:input rdf:ID="manufacture-name">
      <process:parameterType rdf:resource="xsd:string"/>
    </process:input>
  </process:hasInput>
  <process:hasOutput>
    <process:output rdf:ID="pc-parts-model">
      <process:parameterType rdf:resource="pc-parts-configurationA"/>
    </process:output>
  </process:hasOutput>
</process:AtomicProcess>
...
<process:CompositeProcess rdf:ID="finished-pc-configuration">
  <process:composedOf>
    <process:Sequence>
      <process:Sequence>
        <process:components rdf:parseType="Collection">
          <process:AtomicProcess rdf:resource="#pc-parts-configurationA">
          <process:AtomicProcess rdf:resource="#pc-parts-configurationB">
          <process:AtomicProcess rdf:resource="#pc-software-configurationC">
        </process:Sequence>
      </process:Sequence>
    </process:composedOf>
  </process:CompositeProcess>
...

```

Fig. 5. The specified configuration process ontology

We employ the intermediate level ontology as the logical configuration theory which complies with the class, inherit, part, associations, dependencies and different kinds of constraints as basic configuration domain specific modeling concepts. The

ontology is common to all kinds of configuration problems, and provides the upper level knowledge for the specified ontology of PC configuration system.

The ontology includes two views of the same personal computer configuration model. One is the product structure model, which delineates the component parts and relations between them. The other is the function structure model that describes the product or service from the function viewpoint. Using the function structure model, we can provide the satisfactory configuration service solely according to the simple and non-professional function requirement of clients. Afterwards, by using translator these implementation independent models are translated into proprietary knowledge base of problem solving engines, such as Racer.

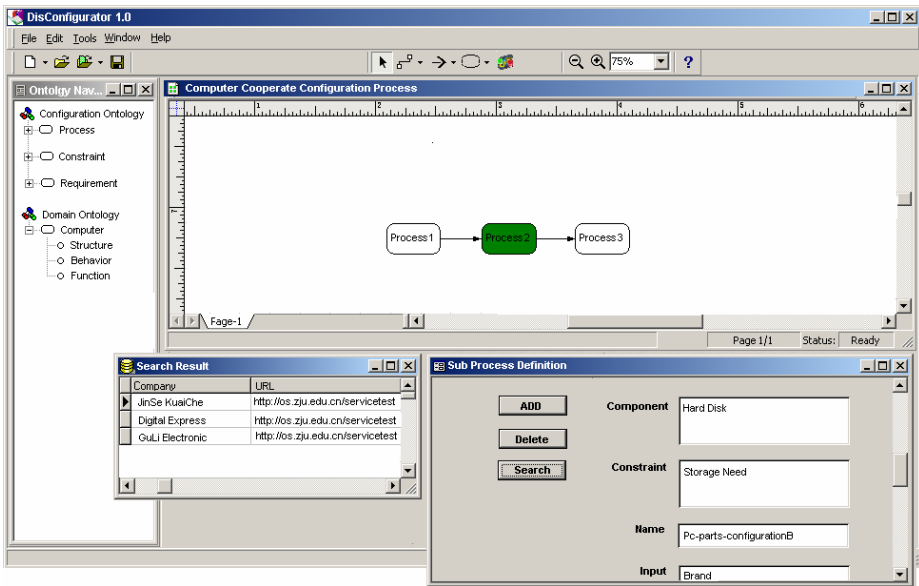


Fig. 6. The interface of the configuration system

Local configuration process ontology offers business the opportunity to reach potential clients like never before through dynamic matching of providers with requestors. Every participant in a dynamic supply chain has its local business policy and plays different roles in different supply chains. For example, company “A” has double roles in a supply chain: one is that it sells finished PCs through their web interface as leader to manage the supply chain; the other one is that it is also a parts supplier. It gets business profit mostly through the distribution of motherboard and CPU. PC configuration service is a way of sales promotion for PC components, so company “A” design two configuration process models. One is for PC part selling, the other for a composing process for describing how company “A” organize a supply chain which will take the benefit of company “A” into consideration first. PC part selling can be modeled as an atomic process that takes as input communication message about the requirement proposed by clients or other cooperators, and replies with a description of the specified type of these components or conflict message. The other one is the composite process ontology. It describes the procedure which is designed by company

“A”. Company “A” takes the whole configuration process divided into three atomic processes, which will be executed by the schedulers corresponding to process control structures.

## 6 Conclusion

Our proposed framework is based on the work of CAWIMOS [6] and takes a further step to take the process knowledge of distributed systems into consideration, which supports the integration among distributed configuration systems. We utilize the Semantic Web service as a platform to obtain the reuse of available configuration problem solving system. This paper demonstrates how to apply Semantic Web technologies to support the integration of configurable products and services in an environment for distributed problem solving. The OWL based configuration service descriptions binding with the current standard are used to build the configuration Web service architecture. However the vision of the semantic is still under development and many details of realization are left for further research. The sponsor of the distributed configuration systems is in charge of the alliance of the supply chain, but how to automatically select partners in eCommerce setting is still an open issue.

## Acknowledgement

This work presented in this paper is supported by the National Natural Science Foundation of China (Grant No. 60174053, 60473052).

## References

1. Pine II, B.J., Victor, B., Boynton, A.C.: Making Mass Customization Work. *Harvard Business Review*, 71 (1993) 108-119
2. McDermott, J.: R1: A Rule-Based Configurer of Computer Systems. *Artificial Intelligence*, 19(1) (1982) 39-88
3. Barker, V.E., O'Connor, D.E., Bachant, J., Soloway, E.: Expert systems for configuration at Digital: XCON and beyond. *Communications of the ACM*, 32(3) (1989) 298-318
4. Sabin, D., Weigel, R.: Product Configuration Framework – A Survey. *IEEE Intelligent Systems, Special Issue on Configuration*, 13(4) (1998) 50-58
5. Haag, A.: Sales Configuration in Business Process. *IEEE Intelligent Systems*, 13(4) (1998) 78-85
6. Yu, B., Skovgaard, H.J.: A configuration tool to increase product competitiveness. *IEEE Intelligent Systems*, 13(4) (1998) 34-41
7. Felfernig, A., Friedrich, G., Jannach, D., and Zanker, M.: Semantic configuration web services in the CAWICMS Project. *International Semantic Web Conference (2002)* 192-205
8. Felfernig, A., Friedrich, G., and Jannach, D.: Conceptual modeling for configuration of mass-customizable products. *Artificial Intelligence in Engineering*, 15(2) (2001) 165-176
9. Zhang, J., Wang, Q., Wan, L., and Zhong, Y.: Product Configuration Modeling Bade on ontology. *Computer Integrated Manufacturing Systems*, 9(5) (2003) 344-351
10. T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2) (1993) 199-220

11. Ankolekar, A., Burstein, M., Hobbs, J.R.: DAML-S: Web Service Description for the Semantic Web. Proceedings of International Semantic Web Conference (2002) 348-363
12. Felfernig, A., Friedrich, G., Jannach, D., and Zanker, M.: Towards Distributed Configuration. Joint German/Austrian Conference on Artificial Intelligence (KI-2001), (2001) 198-212
13. Schlenoff, C., Gruninger, M., Tissot, F., Valois, J., Lubell, J., and Lee, J.: The Process Specification Language (PSL): Overview and version 1.0 specification. NISTIR 6459, National Institute of Standards and Technology, Gaithersburg, MD, (2000)
14. Dean, M., Connolly, D., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., and Stein, L.A.: OWL Web Ontology Language 1.0 Reference. W3C Working Draft 29 July 2002, <http://www.w3.org/TR/owl-ref/>, (2002)
15. Haarslev, V, Möller, R.: RACER: Renamed ABox and Concept Expression Reasoner. <http://www.fh-wedel.de/~mo/racer/>, (2003)

# Online Collaborative Design Within a Web-Enabled Environment

Daizhong Su, Jiansheng Li, and Shuyan Ji

Advanced Design and Manufacturing Engineering Centre, SBE,  
The Nottingham Trent University, UK  
{daizhong.su, jiansheng.li, shuyan.ji2}@ntu.ac.uk  
<http://www.admec.ntu.ac.uk>

**Abstract.** A Web-enabled environment (WEE) for online collaboration has been developed which combines appropriate communications and interconnectivity tools to allow designers to interactively communicate over the Internet in real time, regardless of their IT platforms. The detailed structure of the WEE is presented, followed by description of the key techniques for online collaborative design within the WEE: remote execution of large size executable programs, data file exchange between different CAD systems, and data sharing in real time with distant users.

## 1 Introduction

As the Internet becomes more reliable and widespread, it offers the possibility to enable rapid collaborative product design and manufacture between partners almost regardless of their geographical locations. Much research has been undertaken to achieve this aim using Web-based distributed collaborative environments and related technologies with some, albeit limited, success. Name and Eaglestein listed the tools for a distributed environment [1], Roy et al reported the development of a prototype Web-based collaborative product modeling system [2], Adapalli and Addepalli described different ways of integrating manufacturing process simulations by mean of the Web [3], Kim et al developed a system to store STEP data in an object-oriented database and covert STEP data into VRML data [4], Huang et al studied the Web techniques that can be used for developing collaborative systems [5], Chen et al investigated into a network-supported collaborative design over the Internet/Intranet based on dynamic data exchange [6,7], and Lee et al presented a Web-enabled approach for feature-based modeling in a distributed design environment [8]. The authors' research team has been actively involved in this area, for example, Su et al conducted research in network support for integrated design [9,10,11], and developed a multi-user Internet environment for gear design optimization [12,13,14]; Hull et al developed a software tool for collaborative design and manufacture over the Internet [15].

The processes of design, analysis and manufacture are increasingly requiring very specialised expertise, and may be performed on sites which are distant from each other. The communication of the wealth of relevant engineering information required

between such sites is limited at present in that the data transfer speed can be slow and the hardware and software platforms at each site must be compatible. In order to overcome the limitations, this paper presents a structure for a Web-enabled collaborative environment which combines appropriate communications and interconnectivity tools to allow designers to interactively communicate in real time, regardless of their IT platforms.

Nowadays, product design data is not only managed by the design and production activities [16] but also used in the later stages of the product life cycle [17], which is important to obtain the agility in manufacturing required to improve the competitiveness of companies [18]. The technical data associated with products are different from business data since complex semantics are required, which thus make it very difficult to exchange between different design systems [19].

To meet the challenges mentioned above, a Web-enabled environment and associated techniques have been developed by the authors for geographically dispersed designers to collaborate online. In the following sections, the Web-enabled environment is presented first, followed by two associated techniques: remote execution of large size programs/packages and Web-enable collaborative CAD.

## 2 The Web-Enabled Environment

The Web-enabled environment (WEE) addresses the issue of Internet connections and communications. Its main objective is for the teams/enterprises geographically remotely located to collaborate in design and manufacture with the aid of the Internet and related techniques. Therefore, it is essential to establish an effective means for communication over the Internet. To achieve this, the WEE for collaboration is developed.

### 2.1 Features of the WEE

To develop the WEE, it has to be considered that the partners are not only dispersed geographically but may also work with different platforms, operating systems, protocols and languages. The currently existing ICT tools/systems in this field can address some aspects, such as accessing remote databases, invocation to the remote processes or sharing and integrating of multiple data resources, which are not enough for an integrated heterogeneous environment. As a large heterogeneous platform for collaboration and integration over the Internet, the WEE has the following features:

- (1) Scalability -- The system architecture can accommodate any growth in future load such as new computer processors and/or architectures and tools.
- (2) Openness -- The system can be easily extended and modified. Any new components integrated in the system can communicate and work together with some of components that already exist in the system.
- (3) Heterogeneity -- The system is constructed using different programming languages, operated on different hardware platforms and obeys different protocols. Heterogeneous components have to communicate with each other and be interoperated.

- (4) Resources access and inter-operation -- Resources including software and data should be accessible to, and operated by, all partners.
- (5) Legacy codes reusability -- Existing applications can be integrated seamlessly together with a new application without code-rewriting and can be interoperated with each other. The reusable components enable efficient development processes and reliable application systems.
- (6) Artificial intelligence -- Artificial neural networks, fuzzy logic and genetic algorithms are utilized for the control of process and conflict checking.

The combination of all the above to provide a robust tool for Web-based collaborative environment for design and manufacture is a novel contribution of the research.

### 2.2 Three Tier Architecture of the WEE

The Web-enabled environment presented in this paper is constructed based on the CORBA (common object request broker architecture) technology, because CORBA provides an excellent communication mechanism between client and server. The environment consists of three tiers: A User tier, a Web server tier and an Application tier as shown in Figure 1.

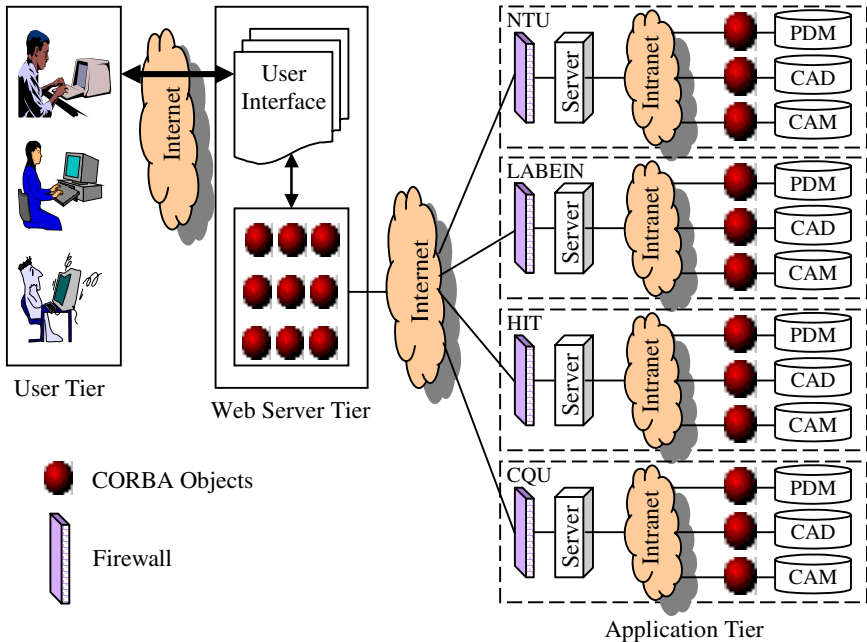


Fig. 1. Architecture of WEE

Each tier has the following functions:

- (1) User tier—Valid customers of the WEE are allowed to visit HTML or XML pages located on the Web server tier over the Internet. The user needs to know



only how to use the graphic interface and does not have to be aware of the technical details behind this.

- (2) Web server tier—This consists of graphical user interface, CORBA agent object and intelligent control of virtual design systems.
  - The user interface includes the whole activities flow according to the design and manufacturing process and Web service tools. All these Web pages contain the inside linkage to the CORBA agent objects.
  - CORBA agent objects are in charge of linking and finding the essential CORBA agent objects associated with the remote applications or software and activating them and retrieving the results. Through CORBA all the components, including applications, software or objects dispersed on the remote heterogeneous systems, can be encapsulated into the objects that CORBA agent objects can manage and find. Therefore the components amongst different partners can be found, shared and interoperated by each other.
  - The artificial intelligent control is being accomplished by employing artificial neural networks, fuzzy logic and genetic algorithms for the intelligent control of the whole process of design and manufacture. It includes the CAD/CAM connection and feature recognition, intelligent distribution with neural networks, intelligent scheduling using GA and virtual and evaluative activities.
  - In addition there are synchronous and non-synchronous communication.
- (3) Application tier—This tier consists of a number of applications and tools needed to implement the design and manufacturing, and services that will be provided by CORBA objects. These applications may be dispersed geographically and also written in different languages, work on different platforms and operating systems, and are encapsulated into components that the CORBA agent can find. The interface of each application, mapped into a CORBA agent object, is separated from the implementation written in respective languages/packages that a partner prefers to use. All these applications consist of different large computing programs, various kinds of PDM, CAD and CAM software, accessing and searching applications for backend database resources such as SQL, DBMS or JDBC, and other resources from customer services.

### 2.3 Development of the WEE

In order to implement the integration and collaboration amongst the partners, the environment has to be provided with the following modules:

- Integration of application
- Communication and interoperation between partners
- Unifying graphical user interface
- Intelligent process management

The user interface is developed mainly in Java combined with other appropriate Web technologies. Applications necessary to the design and manufacture are implemented by all the partners using their own systems, which will ensure the use of heterogeneous systems.

Integration of applications without rewriting legacy codes is one of the key issues on which the environment being developed is based. CORBA is an open, vendor-neutral,

middleware standard that allows the proliferating number of hardware and software applications to communicate with one another using heterogeneous systems. CORBA is a complete distributed object computing framework to extend applications across networks, languages, component boundaries and operating systems. Distributed collaborative systems which are constructed based on CORBA/Java have features such as high-performance, scalability, maturity, inter-operability, support for legacy systems and ease of development. This research utilizes the object technology CORBA, Java and other technologies in the development of the Web-enabled environment.

### 3 Effective Remote-Execution of Large Size Programs/Packages

In order to achieve best product design and low production costs, some large-sized programs, such as design optimisation and finite element analysis software, are often used in the design phase of product development. Usually, such programs/packages are time-consuming in computation and may not be valid to download due to software copyright, or due to their large size and the limited network bandwidth. To remotely execute such software in an effective way in order to conduct on-line collaborative design within the WEE, an approach has been developed which is presented below.

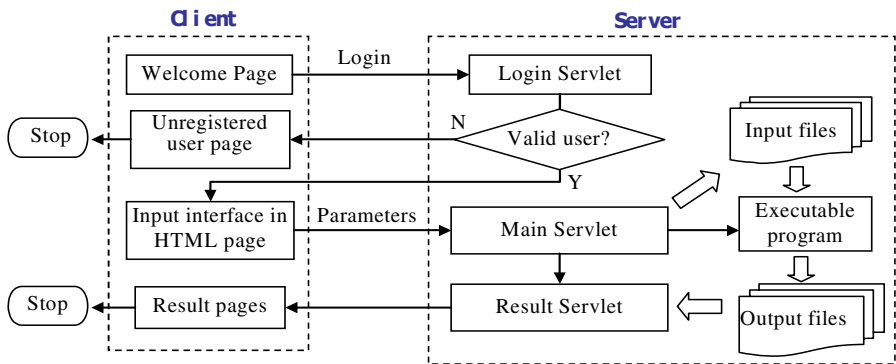


Fig. 2. Remote execution environment based on Java Servlets

#### 3.1 Application of Java Servlets Technology in the System

Java Servlets technology is applied to remotely execute a large sized program over the Internet and to improve the execution speed within the WEE. The executable program is located and executed on the server-side and the results are sent to the client-side after the completion of the program. To accomplish this, a combination of Java Servlets, HTML, JavaScript, Java, HTTP protocol and multi-user environment are utilized.

The structure of the system is shown in Figure 2. The user on the client side inputs the parameters of the program via the input interface in the HTML page. Then these parameters are sent to the server, which calls the Servlets located on the server. When the user clicks on the submit button on the HTML page, a Servlets program is activated by the HTML code. It parses the parameters and writes them into the input files

and invokes the executable program which is located on the server. When the execution is completed and output files are created, Servlets return the results to the client.

### 3.2 Application of CORBA Technology in the System

CORBA technology is used to provide a better solution for this remote execution environment. The CORBA's ORB (Object Request Broker) is utilized as the communication bus for all objects in the system. It enables objects to transparently make requests to---and receive responses from---objects located locally or remotely. The client is not aware of the mechanisms used to communicate with, activate, or store the server objects.

The IDL (Interface Definition Language) is used to define interfaces in CORBA. An IDL interface file describes the data types and methods or operations that a server provides for implementation of a given object. IDL is not a programming language; it describes interfaces only, but has no relation to implementation. The IDL can be mapped to various programming languages, including C, C++, Smalltalk Ada, COBOL and Java.

Figure 3 portrays the architecture of the remote execution environment where CORBA is used as Middleware for the interface between the client GUI written using Java Applet and executable programs such as C/C++ applications distributed on the Web. ORB helps to turn a local application (distributed) into the Web-based object, so that it can be accessed over the network such as the Intranet, Extranet or Internet.

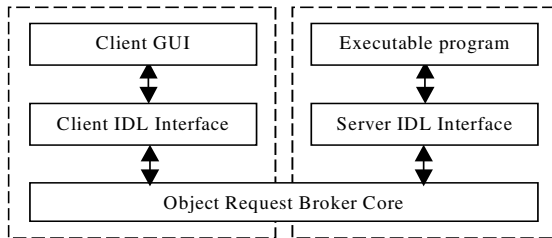


Fig. 3. Architecture of WEE

The client GUI (graphical user interface) captures any user input, and then it uses the CORBA software component, called the ORB, to transparently invoke and pass the parameters to the executable program at the application server. The results of execution are then passed from the server back to the client GUI.

## 4 Web-Based Collaborative Computer Aided Design

In an advanced manufacturing model, all sorts of efficient processes have a close relationship with the geometric model of product, which is the foundation of the operation of an integrated design and manufacturing system. Many commercial CAD and finite element analysis (FEA) software packages, such as Pro-E, UG, SolidWorks, AutoCAD, ANSYS, etc, are widely used in product design and manufacturing.

Although all of these programs support 3-dimensional modeling, the data format is different in each of them. In the distributed collaborative environment considered by this research, a given partner may have different CAD and FEA software from those other partners may have. In order to collaborate with each other, it is necessary to implement robust and reliable methodologies for the exchange and sharing of data between heterogeneous CAD systems.

To achieve this, a neutral file approach for CAD data transfer and a CORBA based approach for CAD data sharing over the Internet are proposed as detailed below.

### 4.1 Neutral File Approach for CAD Data Transfer

Neutral files and neutral file interfaces are needed in order to exchange product data between CAD systems. Direct translators exist but the number of required translators becomes too large if there are many CAD systems involved in the data transfer. For each pair of CAD systems to be able to communicate, two translators are required, one for each direction. For a new additional CAD system, several translators have to be added to each existing CAD system. Figure 4(a) shows the situation using direct translators. When using a neutral file format only one pre- and post-processor is needed for each CAD system. When a new CAD system is added, only one pre- and post-processor needs to be added. Figure 4(b) shows the situation for data transfer using a neutral file format.

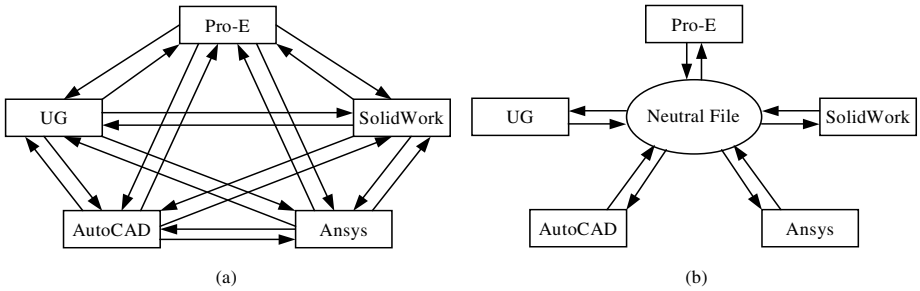


Fig. 4. CAD data transfer: (a) direct data transfer (b) using a neutral file

### 4.2 Web-Based CAD Data Sharing

In order to conduct integrated design over the Internet/Intranet within the Web-enabled environment, it is necessary to provide an online framework to enable geographically dispersed team members to discuss and to modify CAD data simultaneously.

Using currently available commercial Internet communication software, such as Windows Netmeeting, two or more users can share their CAD drawings by remote desktop sharing. However, massive bitmap data needs to be transferred across the Internet, so that the speed of interaction is greatly constrained by the limited network bandwidth. Moreover, such software can only run on the same operating system. To solve this problem, a Web-based approach for collaborative computer aided design

across different operating systems is developed within the WEE. Figure 5 shows the working flow chart for this system.

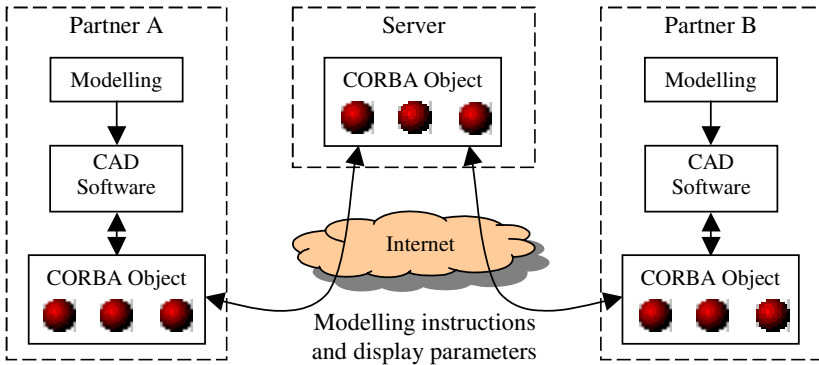


Fig. 5. Flow chart of CAD sharing system

To reduce the load on network transportation, this research puts forward a new approach for real time CAD data sharing. In this new approach, users in different locations start their own process instance with the same CAD software, and connect to the same server by Internet communications that follow CORBA rules. The information transferred through the Internet is not the bitmap data, but the design instructions and display parameters so that the quantity of the Internet transferred data can be greatly reduced and, therefore, the response speed of this proposed collaborative design environment increases manifold.

Assuming partner A is sharing CAD data and collaborating with partner B, when partner A develops a model using a particular CAD software, the modelling instructions and display parameters will be transferred to the server in the meantime, and then, from that server to partner B’s computer and displayed using the same CAD software. When user B works on the model, the process is reversed. In this way, partners A and B can share their CAD data in real time.

## 5 Concluding Remarks

A Web-enabled collaborative environment has been presented in this paper. The three tier structure enables the environment’s advanced features such as scalability, openness, heterogeneity, resources accessibility, legacy codes reusability and artificial intelligence. The key techniques involved in the development of the software environment include CORBA broker for collaboration and remote design using Java Servlets.

The approach developed for remote-execution of large size programs/packages provides an effective tool for remotely accessing software without downloading which is beneficial for utilizing the resources amongst geographically dispersed collaborative teams. In the approach, a combination of Java Servlets, HTML, JavaScript, Java, HTTP protocol and multi-user environment are utilized.

Different from existing systems/methods, the Web-based collaborative CAD presented in this paper utilises neutral files and neutral file interfaces to exchange product data between CAD systems, which has the advantages that only one pre- and post-processor is needed for each CAD system. To reduce the load on network transportation, a new approach for real time CAD data sharing is presented. In the approach, only the design instructions and display parameters are transferred over the Internet, so that the quantity of the Internet transferred data can be greatly reduced and the response speed increases.

## Acknowledgement

This research is supported by the EU Asia IT&C programme (Grant No. ASI/B7-301/3152-099/71553) and Asia-Link programme (Grant No. ASI/B7-301/98/679-023), which has been carried out at The Nottingham Trent University, UK in cooperation with Harbin Institute of Technology and Chongqing University in China, Foundation LABEIN in Spain and Lappeenranta University of Technology in Finland.

## References

1. Name, E.V., Egelstein, G.: The Wired Engineer: Emerging Technologies and the Designer. ANTEC'98 (1998) 3052 - 3055
2. Roy, U., Bharadwaj, B., Kodkani, S., Cargian, M.: Product Development in a Collaborative Design Environment. *Concurr Engng: Res Appl*, Vol.5, No.4 (1997) 347 - 365
3. Adapalli, S., Addepalli, K.: World Wide Web Integration of Manufacturing Process Simulations. *Concurr: Practice Experience*, Vol.9, No.11 (1997) 1341-1350
4. Kim, C.Y., Kim, N., Kim, Y., Kang, S.H., O'Grady, P.: Distributed Concurrent Engineering: Internet-Based Interactive 3-D Dynamic Browsing and Markup of STEP Data. *Concurr Engng: Res Appl*, Vol.6, No.1 (1998) 53 - 70
5. Huang, G.Q., Lee, S.W., Mak, K.L.: Web-Based Product and Process Data Modeling in Concurrent Design for X. *Robotics Comput-Integrated Manufact*, Vol.15 (1999) 53-63
6. Chen, X., Su, D., Li, Z.: Network-Supported Collaborative Design Based On Dynamic Data Exchange. *Proceedings of the International Conference on Computer Aided Industrial Design and Conceptual Design*, 16-20 October 2001, Jinan, China (2001) 448-452
7. Chen, X., Yin, Y., Su, D.: Collaborative Computer Aided Design over the Internet/Intranet. *Proceedings of the International Conference on e-Commerce Engineering: New Challenges for Global Manufacturing in 21st Century*, 16-18 September 2001, Xi'an, China (2001) RID06-1
8. Lee, J.Y., Kim, H., Han, S.B.: Web-Enabled Feature-Based Modeling in a Distributed Design Environment. *CD-ROM Proceedings of the 1999 ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Las Vegas, Nevada (1999)
9. Su, D., Chen, X.: Network Support for Integrated Design. *Integrated Manufacturing Systems - The International Journal of Manufacturing Technology Management*, Vol.14, No.6 (2003) 537-546
10. Su, D., Ji, S., Li, J., Hull, J.B.: Web-enabled Collaborative Environment for Integrated Design and Manufacture. *Proceedings of International Conference on Concurrent Engineering*, 27-31 July 2002, Cranfield, UK (2002) 93-101

11. Su, D., Amin, N., Chen, X., Wang, Y.: Internet/Intranet Based Integrated Design of Mechanical Transmission Systems. 8th Drive and Control Conferences, 13-15 March 2001, London, UK (2001) 13-20
12. Su, D., Ji, S., Amin, N., Hull, J.B.: An Internet-Based System of Gear Design Optimization Using Java Servlets. Proceedings of the International Conference on Computer Aided Industrial Design and Conceptual Design, 16-20 October 2001, Jinan, China (2001) 30-36
13. Su, D., Ji, S., Amin, N., Hull, J.B.: Multi-User Internet Environment for Gear Design Optimisation. Integrated Manufacturing Systems - The International Journal of Manufacturing Technology Management, Vol.14, No.6 (2003) 498-507
14. Amin, N., Su, D.: Enhancement of Speed and Efficiency of an Internet Based Gear Design Optimisation. International Journal of Automotive Technology and Management, Vol.3, No.3/4 (2003) 279-292
15. Hull, J. B., Su, D., Ji, S.: Development of a Powerful Software Tool for Collaborative Design and Manufacture over the Internet. Proceedings of the International Conference on Industrial Tools, 8-12 April 2003, Bled, Slovenia (2003) 399-402
16. Yoo, S.B., Kim, Y.: Web-Based Knowledge Management for Sharing Product Data in Virtual Enterprises. International Journal of Production Economics, Vol.7, No.2 (2002) 173-183
17. Hardwick, M., Spooner, D., Rando, T., Morris, K.C.: Sharing Manufacturing Information in Virtual Enterprises. Communications of the ACM, Vol.39, No.2 (1996) 46-54
18. Gunasekaran, A.: Agile Manufacturing: A Framework for Research and Development. International Journal of Production Economics, Vol.62, No.1/2 (1999) 87-105
19. McKay, A., Bloor, M., Pennington, A.: A Framework for Product Data. IEEE Transactions on Knowledge and Data Engineering, Vol.8, No.5 (1996) 825-838

# C-Superman: A Web-Based Synchronous Collaborative CAD/CAM System

Weiwei Liu, Laishui Zhou, and Haijun Zhuang

CAD/CAM Engineering Research Center,  
Nanjing University of Aeronautics and Astronautics, Nanjing, P.R. China  
liuww9899@sina.com

**Abstract.** Distributed synchronous collaborative design and manufacturing is considered as an important aspect of collaborative design and manufacturing in network environments, and is gaining more and more attention. In this paper, C-Superman, a Web-Based synchronous collaborative system is presented. Then, system implementation mode, system architecture and function modules are described. Furthermore, some key technologies such as session communication and data transmission, concurrent control and consistency maintenance, cooperative awareness in remote environments and load balancing are addressed.

## 1 Introduction

Distributed Collaborative Design (DCD) [1] is a typical application of CSCW in collaborative product design and manufacturing. Participants with various domain expertise collaborate with one another under distributed cooperative working environments established by multimedia computers and telecommunication networks, and accomplish the design and manufacturing of complex products or solve specific engineering problems by means of exchanging information and knowledge, thus meet the requirements of global market [2,3].

Distributed collaborative CAD/CAM is considered as an important development trend of the current CAD/CAM technologies. Distributed collaborative CAD/CAM systems are integrated multi-person-machine-task systems. They can help designers to share design results and cooperate in the same design region so as to undertake design and manufacturing simultaneously and harmoniously. By the working fashions, distributed CAD/CAM systems can be classified into two categories [4]: (1) distributed asynchronous mode; (2) distributed synchronous mode. In the first mode, the coupling relationship among collaborators is loose, i.e., the operation of one collaborator is not transferred to the others immediately but perceived by them after some time. In the second mode, collaborators have close coupling relationship, and real-time communication and cooperation is prerequisite. Therefore the realization of the distributed asynchronous mode is more difficult.

Based on the CAD/CAM system called Superman2000, developed at the CAD/CAM Engineering Research Center of Nanjing University of Aeronautics and Astronautics, the authors implemented a Web-Based synchronous collaborative CAD/CAM prototype system called C-Superman. The rest of this paper is organized



as follows. Section 2 presents the implementation mode of C-Superman. Section 3 briefly describes the architecture and function modules. In Section 4, some key enabling technologies of this system are discussed, such as session communication and data transmission, concurrent control and consistency maintenance, cooperative awareness and load balancing. Section 5 presents the development techniques. Finally, Section 6 concludes the paper.

## 2 System Implementation Mode

Nowadays, under the circumstances of global economy and management, in order to start collaboration work, it is inconvenient to install and configure each collaborator's computer because of the immense fluidity and unpredictability of participants in a design and manufacturing group. Thereby, a three-tiered model based on Browser/Server is adopted in C-Superman. It is a thin client mode [5]. Web browser is the only software required for the client side, which eliminates the demand for installing and maintaining specialized software on each client machine. Though simple and inexpensive, this mode can expand the use of CAD/CAM to many people without additional network facilities. All the tasks for system development, maintenance and upgrade are completed on the server side; hence the overall cost of system implementation is greatly reduced. Collaborators may trigger activities, sub-activities and meta-actions continually through "requests and responses", and then carry out tasks within their privileges to form dynamic action chains expressing design and manufacturing procedures.

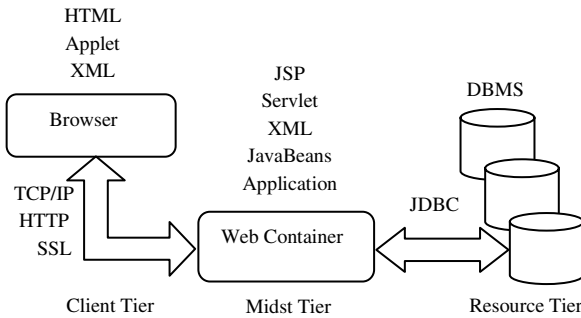


Fig. 1. Implementation mode

In the whole system, communication is the basis, cooperation is the form, and coordination is the core. Some problems like resource and task assignment, secure transmission, and real-time guarantee must be solved. To satisfy the requirements of distribution, opening, safety and platform independence, J2EE (Java2 Enterprise Edition) is introduced to construct the logic frame of C-Superman. J2EE provides the mode of multi-tiered distributed application, component reuse, consistent safety models and convenient transaction control. Thus a scheme of three-tiered Web-centric

application—Client Tier, Midst Tier and Resource Tier, is presented. This can simplify the configuration of client side, solve the problem of its flatness and acquire load balancing via multi-servers. The implementation mode of C-Superman is shown in Figure 1.

Midst Tier is responsible for almost all the application functions of C-Superman, including the generation of dynamic contents, representation and disposal of users' requests, realization of key application functions, implementation of operation rules defined, management of data access and business logics. Resource Tier is a back-end information system that supports data management. Client Tier provides human-machine interfaces.

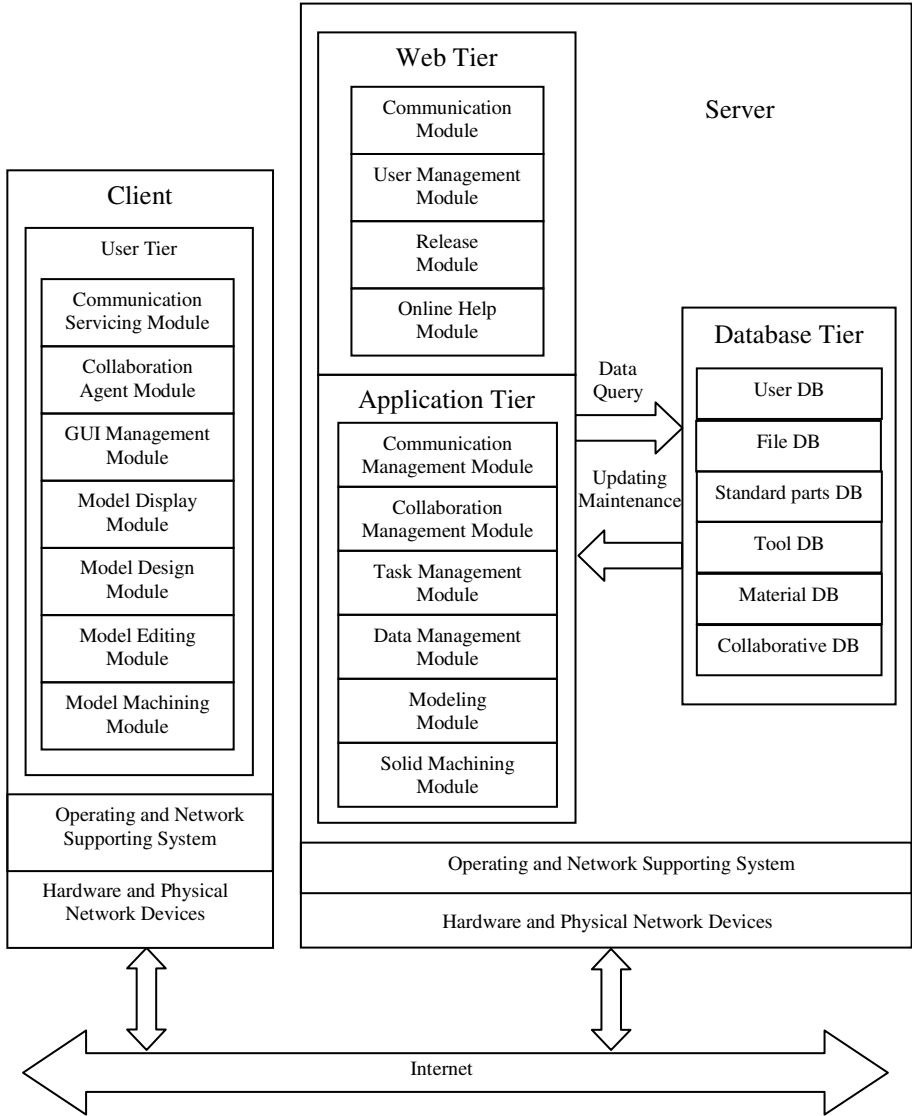
Web Browser can run on many hardware and software platforms, and through Web standards, for instance, HTML, XML, visit the representation logic of Web Container and download an Applet as its GUI (Graphic User Interface) to acquire the operation logic of Web Container. Web Container coordinates simple front-end and complex back-end functions, receives and deals with clients' requests, and cooperates with DBMS via JDBC Pool and distributed transactions. JSP, XML, Servlet, JavaBeans and Application actualize the representation logic and operation logic.

### 3 System Architecture and Function Modules

There are two main approaches for the implementation of distributed CAD/CAM systems, namely concentration and replication. Concentration is a center-controlled architecture following the Client/Server mode. Client side answers for the user's operations and displays models. Yet, shared objects are placed on the server side. Server side generates shared models and broadcasts the results to all collaborators. SPIFF system [6] and NetFeature system [7] are the examples of this kind. Concentration is simple and may keep the information persistent. Furthermore, it is easy for concurrent control, role support and disposal of visiting/accessing authorization. But the problem of network load, delay of communication and bottleneck of the central server also appear when the number of collaborators increases.

Replication is not a center-controlled architecture. In the collaborative group, each client computer has a set of modeling system and shared objects, and can execute the same modeling operation to realize synchronous collaboration. COLLIDE [8] and Co-operative ARCADE [9] are examples of this kind of systems.. The advantages of replication are low network traffic and rapid response. However, its system architecture is complex, and its concurrent control is difficult, especially in heterogeneous environments.

C-Superman adopts the concentration approach, supporting heterogeneous environments and allowing users to join any time. Consequently the problem of consistency and concurrent control is easily solved. As for heavy network traffic, response delay and heavy load balancing, some special strategies will be introduced (for details, see Sections 4.1 and 4.4). Figure 2 illustrates the architecture and function modules of C-Superman.



**Fig. 2.** Architecture and function modules of C-Superman

C-Superman is made up of Client (User Tier) and Server (Web Tier, Application Tier and Database Tier).

The main function modules of User Tier are listed below:

- Communication Servicing Module – finishing the work of sending and receiving data and information to and from Server;
- Collaboration Agent Module – the central module of Client, dealing with the data and information of group awareness, manipulating design objects and cooperating with other modules of Client;

- GUI Management Module – providing the circumstance for the interaction between human and machine, which exempts users from considering whether Server or Client implements the concrete functions;
- Model Display Module – achieving the display of synchronous collaboration to reach WYSIWIS;
- Model Design, Editing and Machining Module – accomplishing the front-end operations of design, editing and machining of 3D geometries.

The main modules of Web Tier are as follows:

- Communication Module – setting up the connection with Client and Application Tier respectively and exchanging data;
- User Management Module – completing the user's registration, logon, authentication, password changing, logout, responsibility assignment, privilege management and the like;
- Release Module – putting out dynamic messages of collaborative groups on the Web to provide references for latter users;
- Online Help Module – giving help information of design and manufacturing on the Web;

Application Tier has the following modules:

- Communication Management Module – coding, decoding, analyzing, reconfiguration, directional transmission of messages, initiative services of Server such as establishment and abolishment of connection;
- Collaboration Management Module – the control center of C-Superman, responsible for the communication transmission for certain modeling task between all the clients of a group, session management, concurrent control and consistency management;
- Task Management Module – assigning individual working space for modeling tasks;
- Data Management Module – linking several different databases and managing them effectively to prevent unlawful and unauthorized operations;
- Modeling Module – with responsibility for specific CAD modeling operations through the popular geometric engine and then scattering complex geometric models into faces and edges to be displayed on Client;
- Solid Machining Module – responsible for CAM operations, for example, 2D contour machining, 3D cavity machining and tool paths calculation of 3D coarse or fine machining;

The modules of Database tier are as follows:

- User Database – recording detailed information and operation privilege of registered users;
- File Database – storing model files, files of tool paths of numeric control machining, log files and so on;
- Standard Parts Database – collection of some standard parts;
- Tool Database – types and parameters of different kinds of tools;
- Material Database – performance and parameters of parts material;

- Collaborative Information Database – recording information like group name, operations of users.

## 4 Key Technologies

### 4.1 Session Communications and Data Transmission

In C-Superman, star topology is introduced to realize the basic communication function. Server is the central node and clients are peripheral nodes. Socket connection based on TCP( Transmission Control Protocol) between server and client is established. It is a bidirectional, ordered flow service without data being repetitive or lost (see Figure 3). Central node achieves communication between clients via message dispatching. For example, in a collaborative group, if Client A wants to send messages to Client B and C, it first sends them to Server by TCP connection between server and itself, then Server relays them to B and C respectively in the same way. Here Server acts as a router. Server makes use of multi-threads, thus can manipulate Clients’ requests simultaneously. For the connection appeal of each client, a thread is assigned by Server to communicate with it.

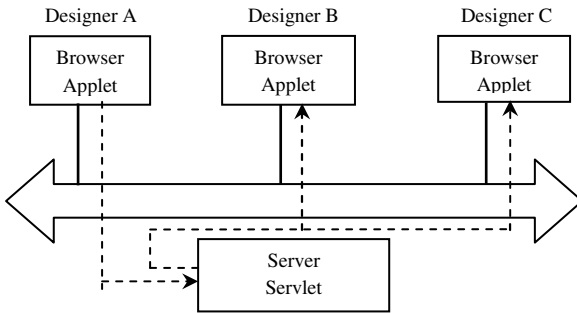


Fig. 3. Principle of basic communication

The amount of data transmitted in C-Superman is heavy due to the concentration architecture. To alleviate the network load and increase cooperation efficiency, meanwhile, guarantee model consistency, operation commands of clients are sorted into network and native commands. Operation commands that will change geometric parameters or topology formats are network commands; otherwise, they are native commands, like browsing or view switching. Network commands are to be sent to server and processed by servicing objects of global information that invoke corresponding modules. Then, processed results will be dispatched to each collaborator. Native commands only affect the view of local users and will be disposed by local servicing objects of message processing.

The online transfers of 3D CAD models might lead to heavy load of system and network. In C-Superman, geometric models, generated by 3D modeling module and solid machining module, are scattered to triangular meshes and transferred to clients for the following reasons: (1) triangular meshes can record the geometric and

topological information of triangular surfaces and have the merits of hardware independence and cross-platform, if drew by Java3D; (2) the display and manipulation of triangular meshes is simple. Moreover, the computation consumption is low. Most hardware supports accelerated drawing of triangular meshes; (3) Effective compressing methods can be taken to improve transmission rates of triangular meshes. Network commands are coded as Command / Parameter, which may reduce information flow significantly.

One possible compression way for geometric data sent back to clients is to extract information of triangular meshes and scattered points. Each triangle is expressed by 9 coordinates of 3 points. If float type is used, one triangle needs 72 bytes, of which 36 depicts vertices and the other for normal vectors, without considering attributes of color and textures. This method makes each vertex depicted 6 times, which consumes large storage space and increases transport time of modules. One improved solution is to express triangles and vertices separately to set up Vertex Spanning Tree and Index Sequence. 3 coordinates express each vertex while Triangle Description Table expresses the triangles. Each table item only includes three sequence numbers of one triangle. Besides, GZipInputStream and GZipOutputStream are used to compress data further. So, the data transferred are reduced greatly and system responses more quickly.

## 4.2 Concurrent Control and Consistency Maintenance

Floor control is used in C-Superman. Before sending network commands, collaborators must apply for and obtain Floor. The operations changing geometric models by collaborators without Floor will be shielded. Hence their operations will not be overlapping so that object handles are consistent.

The mechanism of arbitration is that one collaborator applies for Floor and then Chairman sanctions it or not. The establisher of the collaborative group is Chairman entitled to the right of sending network commands. If a collaborator wants to modify models, he must request Floor and wait for Chairman to authorize. Once getting the Floor, he will replace the latter as Chairman. Accordingly, only one collaborator that owns Floor, namely Chairman, exists at a time. By this way, conflicts of concurrent operations are avoided.

## 4.3 Collaborative Awareness

Besides the online text discussion, in C-Superman, special real-time audio and video subsystem is developed to promote the interaction among collaborators and improve cooperative efficiency.

This subsystem is implemented by JMF (Java Media Frame), which utilizes RTP (Real-time Transport Protocol). Detailed discussions are beyond the scope of this paper. Here are the general procedures: CaptureDeviceManager firstly judges audio and video capturing devices that collaborators own such as microphone, video camera, then initializes them. Secondly, Processor is constituted and output stream is obtained. RTPManager transfers video and audio data to Multimedia Server. Thirdly, Multimedia Server mixes the audio and video data to construct RTP streams and broadcasts them to all collaborators. Finally, collaborators receive RTP streams, separate audio and video streams, and play via their own Processors.

For the transmission of audio and video data, real time response is more important than reliability. So IP multicast is used to save network resources and improve the efficiency of network transfers. In addition, default security model does not permit the client interface – Applet, to use multimedia-capturing devices. To solve this problem, Digit Signature is made to acquire the right of using local audio and video devices.

#### 4.4 Load Balancing

The realization of dynamic load balancing mostly resorts to hardware. To small or middle-sized datacenters, the cost of buying and maintaining such hardware is expensive. Though agile for control, traditional approaches of load balancing have the following shortcomings: (1) special devices like routers are needed to manipulate request/response messages between client and server; (2) cooperative servers must lie in the same LAN or domain; (3) configuration is unavoidable when adding a new server.

The main idea of load balancing in C-Superman is to apply several servers to realize dynamic configuration of client accesses [10]. One server provides a public entrance to all the clients by which users visit Server. Servlet running on the public entrance acts as a reference point of load balancing. According to dispatching conditions, users' accessing requests are automatically repositioned to other assistant servers based on Servlet's characteristic of positioning. Later the assistant server communicates with the corresponding client. This approach only requires a CAD/CAM application copy on each server to interact with GUI downloaded to the client.

The advantages of this approach are: (1) pure software-based technology with no need of special hardware to manipulate messages between client and server; (2) cooperative servers may be located in different LANs or domains, thereby share load at different networks; (3) it is easy to add a server without complicated configurations, so it will not affect existing servers.

## 5 System Development Technologies

The selection of development languages of C-Superman must consider function, speed and cross-platform operations. Geometric modeling and transformation that emphasizes speed comparatively, is the primary job. C/C++ has advantages in this aspect. On the other hand, Java is platform-independent and suitable for developing client side modules. Consequently, based upon ACIS - a solid modeling kernel, C++ is used to realize complex geometric modeling modules; JSP, Servlet and JavaBeans as toolkits for Web Server; Java and Java3D for the development of GUIs; JMF for the implementation of multimedia subsystem. JBuilder 7 and Visual C++ are the main IDEs for C-Superman implementation. Web Server Container adopts Apache Web Server. BEA Weblogic is used for Application Server Container. Database adopts Oracle. Microsoft IE or Netscape is the only requirement for clients.

## 6 Conclusion

This paper describes the implementation mode, architecture and main function modules of C-Superman. It also discusses the key implementation technologies including

session communication and data transmission, concurrent control and consistency maintenance, cooperative awareness and load balancing. This C-Superman system has been validated on the Internet. The research of this project will promote the evolution of application modes of CAD/CAM systems. Predictably, with the solving of network bandwidth and delay problem, Web-Based synchronous collaborative CAD/CAM systems may have a splendid future.

## Acknowledgements

The authors would like to express their sincere thanks for financial support by Encouragement Project of Excellent Youth Teachers of Higher Educational Institutions of Ministry of Education and Funds of Youth Science and Technology of Jiangsu Province of P.R. China under the project code: BQ2000004.

## References

1. Wang, G., Xu, G.: Structure Model of Supporting Platform for CSCW. *Chinese J. Computers*, 20(7) (1997) 718-724
2. Chen, L., Song, Z.J., Liavas, B.: Exploration of A Multi-User Collaboration Assembly Environment on The Internet: A case Study. *Proceedings of the ASME Design Technical Conferences and Computers and Information in Engineering Conference, DETC/CIE-21291*, Pittsburgh, Pennsylvania, September (2001)
3. Rezayat, M.: The Enterprise-Web Portal for Life-cycle Support. *Computer-Aided Design*, 32(2) (2000) 85-96
4. Maher, M.L., Rutherford, J.H.: A Model for Synchronous Collaborative Design Using CAD and Database Management. *Research in Engineering Design*, 9(7) (1997) 95-98
5. Serbedzua, N.B.: Web Computing Framework. In: *Journal of System Architecture*, 45 (1999) 1293-1306
6. Bidarra, R., Van den Berg, E., Bronsvort, W.F.: Collaborative Modeling with Features. *CD-ROM Proceedings of 2001 ASME Design Engineering Technical Conference*, Pittsburgh, Pennsylvania, DETC2001/CIE-21286 (2001)
7. Lee, J.Y., Kim, H., Han, S.B., Park, S.B.: Network-centric Feature-based Modeling. *Proceedings of Pacific Graphics '99 — The Seventh Pacific Conference on Computer Graphics and Applications*, Los Alimitos, CA. IEEE Computer Society. (1999) 280-289
8. Nam, T.J., Wright, D.K.: CollIDE: A Shared 3D Workspace for CAD. *Proceedings of 4th International Conference on Networking Entities NETIES98*, Leeds, 1998
9. Stork, A., Jasnoch, U.: A Collaborative Engineering Environment. *Proceedings of Team-CAD'97 Workshop on Collaborative Design*, Atlanta, US, (1997) 25-33
10. Liu, W., Zhou, L.: A New Approach to Load Balancing in Web-based CAD/CAM System. *Mechanical Science and Technology*, 22(5) (2003) 842-844



# Developing a Multidisciplinary Approach of Concurrent Engineering

Heming Zhang<sup>1</sup> and David Chen<sup>2,\*</sup>

<sup>1</sup>Department of Automation, Tsinghua University,  
Beijing 100084, China  
hmz@cims.tsinghua.edu.cn

<sup>2</sup>LAPS, University Bordeaux 1, 351, Cours de la liberation,  
33405 Talence, France  
chen@laps.u-bordeaux1.fr

**Abstract.** This paper presents an industrial application to implement concurrent engineering in a railway rolling stock manufacturer in China. To extend the application towards a knowledge-based distributed collaborative design system, some theoretical thoughts and conceptual considerations about a human centred approach based on knowledge interactions between stakeholders and learning are outlined. Future relevant research to develop multidisciplinary design approach is also discussed.

## 1 Introduction

Design is multidisciplinary in nature. Considering design globally and taking various aspects into account is a necessity in the current industrial context to develop better products. This multidisciplinary character has at least two dimensions. On the one hand, the development of design models and methodologies must take into account the achievements of various scientific disciplines such as engineering science, cognitive psychologies, mathematics, cybernetics, and social sciences. On the other hand, the design of complex products has the character of multidisciplinary collaboration between various engineering domains (e.g., mechanical, electronics and electro-techniques). Moreover, the implementation of complex product development system is concerned with information integration, process reengineering and optimization, integrated product teams organization and management, resource optimisation, etc.

### 1.1 Context of the Work

In recent years, an important project is in preparation in China: the construction of Beijing-Shanghai TGV railway. No one doubts the highest profile of this project under way in China because it will directly affect other high-speed railways that will be constructed in China in one or two decades. The competition is keen among the French TGV, the Germany magnetic-levitation trains and the Japanese bullet trains.

---

\* Corresponding author.

Within this context, the development of rolling stocks is seen as a key technology for the successful implementation of the Beijing-Shanghai high-speed line. In the long term, the Chinese rolling stock enterprises will develop high-speed rolling stocks by combining the imported technology and the fruits of domestic research and development.

Consequently, there is a strong need from Chinese locomotive and rolling stock industry to shorten product engineering process, improve locomotive and rolling stock designs, and capitalize knowledge. To reach this goal, it is necessary to develop Concurrent Engineering and collaborative design in distributed networking environments. It is also necessary to solve some theoretical problems in order to better understand the design process, correctly capture, represent and manage design knowledge, and identify interactions among all actors involved in such a complex design project.

## 1.2 Concurrent and Collaborative Design Approach

Concurrent engineering (CE) concept was first introduced in 1980's by DARPA (Defence Advanced Research Projects Agency) in US to shorten the product development process. According to Ulrich and Eppinger [13], the product development process can be described as "the sequence or steps that an enterprise employs to conceive, design and commercialise a product". The traditional, sequential, and problem-oriented approach prescribes a logical cause effect relationship between current design problems and solutions. There are numerous frameworks and approaches related to concurrent engineering (e.g., approaches presented in [10] and [11]). Most of these approaches deal with how CE is done internally to a company. Some also deals with the roles of suppliers in CE environment (e.g. Fleischer and Liker's approach [12]). Most of these approaches focus on the use of information technologies to realise integration of various tools and on the process reorganisation/optimisation. Human aspects and knowledge capitalisation are still a weak point.

In summary, the characteristics of concurrent engineering and multidisciplinary collaboration approaches for complex product developments can be stated as follows:

- As a complex product usually consists of many subsystems, such as electronic, mechanical, control, and software, various CAX/DFx tools of the different disciplines are used in the design and simulation activities to support the product development in the process of product development lifecycle.
- The distributed, heterogeneous digital models and product data generated by the tools of different disciplines need to be managed and integrated. A sophisticated management technique, including the management and optimization of diverse data, models, tools, IPTs (integrated product teams) and processes, needs to be implemented as well.
- A product data management or PDM based collaborative platform efficiently manages the product data, simulation models and related processes, and integrates the various CAX/DFx tools.

## 1.3 Purpose of the Paper

This paper first presents an industrial application concerning the implementation of a concurrent engineering approach in Qiqihaer Railway Rolling Stocks (QRRS) Ltd.

Co. The proposed approach mainly focuses on the technical and technological aspects of complex product development in the company. Based on the experience gained and requirements/problems identified during the implementation, a potential extension towards a knowledge-based distributed collaborative design system is outlined. Consequently the second part of the paper discusses a knowledge interaction model placing 'design as leaning' at the centre of the design knowledge management system. The paper is structured as follow. After the introduction, Section 2 presents the as-is situation and user requirements of QRRS as well as the implemented concurrent engineering system. In Section 3, some potential developments towards a knowledge-based distributed collaborative design system are discussed with a focus on some basic concepts and principles. Human oriented approach based on knowledge interactions between stakeholders and learning is proposed. Recommendations on some theoretical research are given in Section 4. The last section concludes the paper.

## 2 Concurrent Engineering for the Rolling Stocks Development

Qiqihaer Railway Rolling Stocks (QRRS) Ltd. Co. is one of the major railway rolling stock manufacturing enterprises in Asia. This section briefly reports the experimentation of concurrent engineering in QRRS.

### 2.1 The As-Is Situation of QRRS

In QRRS, the product development process still follows a traditional approach. It proceeds in serial processes and leads to delay in the information feedback of downstream to up-stream. The different stages cannot efficiently share and exchange the information, as well as coordinate for design modifications. The various design tasks are carried out in an isolated way, lacking of the enabling tools to consider the various factors of process planning, manufacturing, assembly in the early design stages. Only when one stage is finished can the next stage start. Design errors are often found in later stages of the design, sometimes even during manufacturing. Thus occurs the long cycle: design-manufacturing-design modification. This sequential workflow cannot satisfy the requirements of new product developments.

In summary the hierarchical and department-isolated organization and the sequential product development process lead to a lot of problems:

- The communication among different designers is seriously obstructed. The product information cannot be effectively integrated in the different design stages.
- The cycle of production preparation is prolonged, and the design modification request is frequently generated. Some insignificant activities still exist in the process of product development.
- It is difficult to manage and maintain the tremendous amount of engineering documents and product data, and this brings the severe losses because of inconsistent versions of product data.

### 2.2 User Requirements

Generally speaking, there is a strong demand from Chinese locomotive and rolling stock enterprises to improve the current design workflow using more computer aided

tools in product design, engineering analysis, process planning and NC machining. Concurrent engineering is seen as a way to change (with some limits) serial processes into concurrent processes. The capabilities of manufacturing, assembling and testing must be considered in the early processes of product development, to reduce unnecessary modifications and to attain product design in one-time success.

More particularly, consolidated requirements of QRRS are summarised as follows [8]:

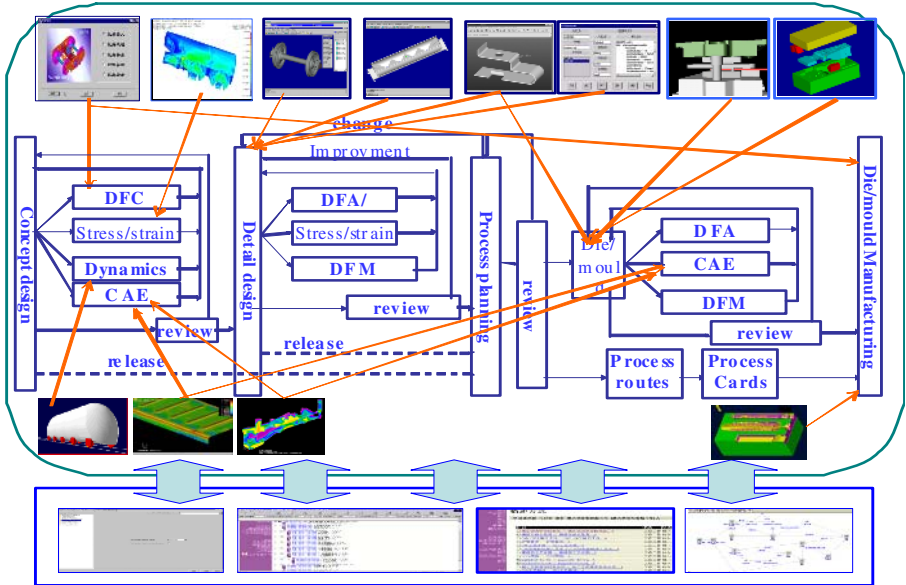
- Organizing the integrated product teams (IPTs) to support teamwork and concurrent tasks (The IPTs organization and management being quite different from the traditional ones).
- Enabling technologies and tools of concurrent engineering, i.e., DfX (DfA: Design for Assembly, DfM: Design for Manufacturing) which help making decisions in the early design stage, must be based on CAx (CAD, CAE, CAPP, CAM, etc.), and integrating CAD/CAM.
- Establishing PDM and collaborative design environment based on computer network, to support the inter-operations, the evaluation and modification of the design simultaneously under heterogeneous systems, and to support the distributed IPTs to work collaboratively and to share product data timely.
- A concurrent process model different from former design and manufacturing process must be established. Through pre-release and design review, designers from related disciplines should take part in the design earlier, discover design errors ahead of time and remove potential problems in the early stages.

### 2.3 Concurrent Engineering Implementation

The traditional development processes of QRRS were reengineered and a large number of design iterations and work-over are avoided. The improved QRRS product development process is shown in Fig. 1. At early stage of product development, manufacturability of sheet metal, structure intensity, stiffness and dynamics performance of railway rolling stocks are comprehensively considered to reduce design errors. Meanwhile, by adopting DfX, various factors of product manufacturability, assembly and techniques can be considered in product design stage, thereby probability of one time design success is increased. Windchill-based PDM has been implemented as the concurrent design framework of the railway rolling stocks and locomotives product development. Hereinto basic environment management, encapsulation and integration of application tools, document management/electronic vault, management of concurrent product development process and workflow four modules were realized through function configuration or application development based on Windchill framework. Product data management system provides not only transparent collaborative network between IPTs in distributed and heterogeneous environment and product lifecycle management, but also orderly management of all the data relevant to products, to ensure that right information be delivered to right person at right time and in a right format.

Simulation toolkits are the kernel technology of the new Rolling Stock and Locomotive system. They provide the technologies and methods of quick finite element analysis, structure optimization and dynamic modification. Adopted some commercial software, many simulation applications were realized in the process of Rolling Stock

and Locomotive development, such as mechanical quality simulation of bodywork and key accessories, oscillation and impulsion response simulation, walk dynamics simulation, and brake system simulation. All the simulation applications can be integrated in the PDM system and the collaborative simulation platform [9].



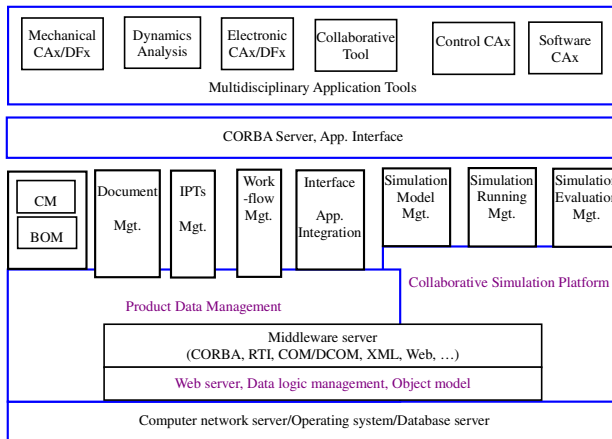
**Fig. 1.** The Rolling Stock and Locomotive Application of Concurrent Engineering

More precisely the process reengineering implementation in QRRS is concerned with the following aspects:

- 3D digital product design: The product design and downstream design of casting mould, stamping tool and fixture equipment of railway rolling stocks were realised from 2D CAD tool to 3D CAD tool. The digital prototype is accomplished. The 3D digital models were available and shared to the related designers.
- Analysis and simulation of the 3D models: Based on the 3D product models, the structure intensity analysis, stiffness analysis, dynamic performance analysis, kinematics simulation and casting process analysis of the railway rolling stocks were also realised.
- A Cost Estimation System (CES) was developed and implemented to quote timely. It is based on the result of database of the cost analysis of product, especially series products. Designers and decision makers can browse the analysis results and make decisions accordingly. PDM system provides the information of parts and components, which can also be directly provided by CAD/CAM/CAPP systems. This information is then directly stored in the components database. Cost management information system, which belongs to the QQRS ERP system, updates the basic cost database (including parts and components cost, man-hour

of work orders). Estimation results are returned to design department through PDM platform.

The architecture of the complex product development environment to meet multidisciplinary collaboration and concurrent engineering is shown in Fig.2. It contains four sub-systems: (1) Management and collaboration sub-system provides project management, product development process management, workflow management, and product models and data management related to the concurrent and collaborative product development lifecycle. (2) Collaborative simulation platform is mainly based on the shared data management of PDM. It provides a collaborative simulation environment and efficiently integrates the simulation modeling, running and evaluating tools. Through computer network, it connects IPTs that possess all kinds of data and supports their cooperative work and innovation. (3) The sub-system of the multidisciplinary application tools consists of the modeling and analyzing application environment, such as mechanical CAx/DFx tools, the kinematics/kinetics modeling and simulation tools, electronic CAx/DFx tools, control CAx tools, and visualization environment. For example, DFA and DFM tools can apply to analyze the assemblability and manufacturability of product digital models, comprehensively considering various factors in the early stage of product development. (4) Supporting environment sub-system comprises PDM, database and Internet-based environment. It is the supporting platform and the basis of the collaborative design environment of complex products.



**Fig. 2.** Architecture of Concurrent Product Engineering Environment

The collaborative design environment uses CORBA and Web server to implement the multidisciplinary CAx/DFx tools calling, exchanging and sharing the information and resources. In this distributed information integration and sharing application service mode, different application servers are encapsulated according to the CORBA servers. The Web servers are also established according to the CORBA sever mode. The Web server provides the links that the users can access the server by using client software or any standard browser.

## 2.4 Remarks

The implementation of concurrent engineering in QRRS is considered as a first step towards a knowledge-based distributed collaborative design system. The focus was on information sharing and integration using computer technologies. It allows shortening significantly the product development delay. However, some problems remain to be solved and challenges to be addressed. For example, how to capitalise, manage and maintain design knowledge which is dispersed and fragmented in a big company like QRRS? How to improve knowledge collaboration between all actors involved in the product development lifecycle? Concerning the design itself, some theoretical problems also need to be studied in order to better understand the design process and promote repeatable and verifiable scientific design approach.

## 3 Potential Further Development of Collaborative Design

This section discusses the basic concepts of extending the concurrent engineering system to a distributed knowledge-based design system in QRRS.

### 3.1 Commissioned Design

Implementing distributed and knowledge-based design system implies the identification of various stakeholders involved and knowledge interactions between them. The design considered in QRRS is the *commissioned design* system which is supposed to be the most complete form. It refers to a specific set of activities pertaining to a project that is initiated by a sponsor. The sponsor may differ as compared to the users and other stakeholders of the system where the designer has to cope with:

- the possibly conflicting requirements of the stakeholders,
- the constraints imposed by the availability of the components and technologies,
- the constraints imposed by the capabilities of the production (and implementation) system [4].

*Remark 1.* Very often the commissioned design is *collective design*. Collective design situations are of two types [2]: *distributed design* and *co-design*. In distributed design situations, the actors of the design are simultaneously, but not together, involved on the same collective process. However, in co-design, “design partners develop the situation together: they share an identical goal and contribute to reach it through their specific competence; they do this with very strong constraints of direct co-operation in order to guarantee the success of the problem resolution” [2].

*Remark 2.* Collective design is distinguished from the *participatory design*. “Participatory design is the direct inclusion of users (and all other stakeholders) within a development team, such that they actively help in setting design goals and planning prototypes. It emphasises that designers must deeply understand the human activity systems that will be affected by their designs”. The difficulty of this approach is to define the degree of the participation. There has to be some limit, because the user and stakeholder should not take over the design task.

### 3.2 Knowledge Interaction Model

The commissioned design studied from social science point of view can be modelled as interactions between people. The interactions are concerned [4] [1]:

- With the co-designers, one of them may be the 'chief' designer who controls the design process,
- With the sponsor, i.e., the one who defines the objectives of the project of which the design process is a part (the stakeholder can delegate part of his/her authority to the project leader),
- With other stakeholders generating requirements to the result of the design process: users, people in charge of engineering, production, implementation of the artefacts as well as those in charge of operating the artefact, those in charge of maintenance and management, and finally, those in charge of laying down or breaking up the artefact once its technical or economical life-time is completed.

The possible stakeholders to consider are sponsors, users, design managers, designers, colleagues (designers), production engineers, and maintenance engineers. The processes are: expression of mental processes and constructed by a designer, storage in a system, inter-personal interactions (one to one), group interactions, and corporate level interactions. The objects (documents, either real or virtual/computer-based) are: requirements, design specifications, design standards, design methods. The sequence of contexts in which interactions is to be analysed is: (1) the designer as such (with pencil and paper), (2) the designer with his tools, (3) the designer and the co-designer, (4) the designer and the user, (5) the designer and the post-design stakeholders (people responsible for production, implementation and maintenance, etc.).

The rationale is considered as follows [4][1]. (1) Design processes are derived from scientific problem-solving processes. (2) The solution of the problem is: (i) in the design term, the product design, i.e., definition, specification, drawings and the (process) plans for the next phases in the project, and (ii) in the project term, the product itself. This leads to the solution repository. (3) The baseline (growing overtime) of the problem-solving process consists of: (i) the problems encountered by the stakeholders at the current situations (as-is), and (ii) their needs. This leads to the problem repository. (3) The process is carried out using design resources, i.e., methodologies, specific methods, techniques, templates and tools, information resources and expertise. This leads to the design resources repository. (4) The process involves learning, i.e., meta-knowledge about particular process instances such as lessons learned, problems encountered, hypothesis formulated and design decisions made. This leads to the design process repository. (5) The design process is to be managed (micro-management of the design process itself and macro-management at the level of project) through objectives, schedules, milestones, specific constraints, tracking reports and progress evaluations. This leads to process control repository.

### 3.3 Learning Centred Design Model

The design modelled from the problem-solving point of view involves necessarily learning which is another important dimension of the product design. The collective design approach should explicitly describe the learning process in the context of



design (knowledge and expertise as the result of the process). There are different leaning cycles: long, medium and short terms. Short term learning is more related to solving technical problems in a particular design process. Medium term learning, besides of technical learning, is concerned with design management, co-ordination and planning related issues. Long term learning (beyond the design and manufacturing of a class of objects with one or multiple instances), refers to consolidate and capitalise knowledge learned at short and medium terms. There are also various entities of learners: individuals, groups and organisations. Design knowledge management deals with the way of managing efficiently the results of these various learning cycles. Fig. 3 shows the links between the core design process, the design process control, project management and learning [1].

At the macro level, design can be defined from the problem-solving point of view and described as a structured set of steps to follow (i.e., a systematic approach) with the iterations between them. The model is a simplified and aggregated representation of all possible processes of designing. It distinguishes: (i) the core processes of design, (ii) the interactions with other designers and stakeholders in the case where different people are involved in the design project.

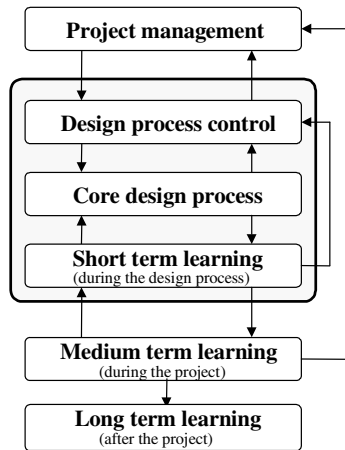


Fig. 3. Partial representation of the design process model (from problem-solving point of view)

#### 4 Recommendation for Some Theoretical Research

There exists an important gap between theory and industrial applications. On the one hand, engineering design performed in enterprises is not based on a theoretical foundation; the design result varies according to experience of the designer. On the other hand, some existing design theories are not used in industry. Most of the design theory developments were carried out on the basis of a single scientific disciplinary. They mainly focus on technical and technological aspects of design. A multidisciplinary design theory is still missing.

#### 4.1 Multidisciplinary Framework for Design

To develop multidisciplinary collaborative design, it is necessary to identify possible contributions from different scientific disciplines (philosophy of science, engineering sciences, cognitive and social sciences) and structure them into one consistent framework. For example, the Philosophy of Science brings some epistemological clarification [5]. The traditional approach considers that a theory has to be testable and truth oriented. The model-theoretical approaches are best characterised by technology-oriented action-theory and construction procedures. The systematic approach [6], AI-based and case-based designs fit better within this frame. The systematic approach has been developed within the frame of Engineering Science. It believes that there exist a finite number of steps, i.e., a structured procedure to follow to obtain a design solution. However, from Cognitive Science point of view, design activity is organised in an “opportunistic” way because of “cognitive cost” [7]. As a consequence a “helpful design theory will support human design problem solving only if it will not force a strictly systematic procedure following theoretically optimal phases and steps” [3]. A realistic design approach should provide the main steps of design process and allow at the same time opportunistic processing.

To better understand what designers do when they work together, Social Science allows knowing how an organisation produces good designs and which organisations cannot produce good designs. The techniques used to enhance team-work (like project management, structured meetings, negotiation, room layout, and workplace ergonomics) are process and management-centred design techniques rather than artefact-centred. Social, environmental and ethical factors, like technical factors or economic factors have influences on designing.

Moreover, the cognitive approach seems to provide the best hope for a better understanding of design process in the case of mono-designer (i.e., the mental knowledge manipulation). In the collective design situations, social science approach which views design as a set of interactions between people (not only designers but also users and stakeholders), and between humans and computer tools, constitutes a complementary dimension to other approaches. However, there is no evident reason not to consider that the collective design is mono-designer based plus some specific activities such as coordination, communication, synchronisation and conflicts solving [7].

#### 4.2 Ontology Modelling Technique

The engineering design can be defined as the evolution of a flow of information representing the product knowledge. Thus it is necessary to investigate the use of ontology in the development of design information systems. Ontology is the formal method for descriptions of shared knowledge in a domain. Sharing and reuse of ontologies across different domains and applications can therefore improve the design. The use of ontology modelling technique can also support the development of interoperability of various design systems and tools. It is necessary to study the usability and reusability of ontologies by construction and validation of an ontology for a large and complex domain, and capture meta-level and tacit background knowledge. The ontology consists of the three engineering ontologies formalizing the three viewpoints on physical devices. These viewpoints themselves are constructed from smaller abstract

ontologies. The interdependencies between these ontologies are formalized as ontology projections. This gives us a set of ontologies of varying genericity and abstractness. Identifying these separate ontologies not only makes it easier to understand the domain because classes and ontological commitments are added incrementally, it also increases the ability to share and reuse parts of knowledge.

## 5 Conclusions

The implementation of concurrent engineering in QRRS is a first step to transform the Chinese traditional product development model to a multidisciplinary collaborative design approach. It is not only necessary to consider the collaboration between various engineering disciplines such as mechanical Engineering, electronics, and technology integration, but also other scientific disciplines in particular, social sciences (including cognitive psychology). The findings from design theories should also be taken into account in developing practical methodology. In this paper, based on the concurrent engineering experimentation in QRRS, a human centred knowledge interaction based distributed design system is outlined. Future work is concerned with the description of knowledge objects exchanged between stakeholders/actors through various knowledge repositories on the one hand, and on the other hand the interaction mechanisms between them.

## Acknowledgement

The application reported in this paper is supported by the National Natural Science Foundation of China (NSFC 69884002) and CIMS/863 Program of China High-Tech Plan (2002AA411320). The authors also thank Prof. Xiong Guangleng, Dr. Fan Wenhui and the CE team for the implementation of concurrent engineering in QRRS.

## References

1. Chen, D.: Developing a Theory of Design through a Multidisciplinary Approach, Proc. of IEEE Conference on Systems, Man and Cybernetics, Hammamet, (2002)
2. Darses, F., Détienne, F., Falzon, P. and Visser, W.: A Method for Analysing Collective Design Processes, Research Report, n°4258, INRIA, (2001)
3. Hacker, W.: Psychological contributions to and demands on a General Theory of Design, Proc. of the workshop on Universal Design Theory, Grabowski, H., Rude, S. and Grein, G., (Eds.), Shaker Verlag, (1998)
4. Huysentruyt, J: Research in design theory, Research Note, version 1.0, (2002)
5. Lenk, H.: Epistemological remarks concerning the concepts 'theory' and 'theoretical concepts', Proc. of the workshop on Universal Design Theory, Grabowski, H., Rude, S. and Grein, G., (Eds.), Shaker Verlag, (1998)
6. Pahl, G. and Beltz, W.: Engineering Design - A Systematic Approach, Second edition, Ken Wallace (Ed.), Springer-Verlag, London, (1996)
7. Visser, W: Individual and Collective Design: The Cognitive - Ergonomic Approach, Research Report, n°4257, INRIA, (2001)

8. Zhang, H.M., Xiong, G.L. and Li, B.H.: Study on Process Reengineering and Integrated Enabling Tools of Concurrent Engineering, Proceedings of 2002 International Conference on Concurrent Engineering, Cranfield, UK, (2002)
9. Fan, W.H., Xiong, G.L. *et al.*: Enabling Technology and Tools of Railway Rolling Stocks Product Concurrent Development, Computer Integrated Manufacturing Systems [China], 8(7) (2001) 54-58
10. Prasad, B.: Concurrent Engineering Fundamentals: Integrated Product and Process Organisation, Upper Saddle River, NJ: Prentice Hall PTR, (1996)
11. Hull, F., Collins, P. and Liker, J.K.: Composite forms of Organisation as a Strategy for Concurrent Engineering Effectiveness, IEEE Transactions in Engineering Management, 43(2) (1996) 132-142
12. Fleischer, M. and Liker, J.: Concurrent Engineering effectiveness: integrating Product development across organisations, Cincinnati, OH: Hanser-Gardner, (1977)
13. Ulrich, K. and Eppinger, S.: Product Design and Development, McGraw-Hill, (2000)

# Hardware/Software Co-design Environment for Hierarchical Platform-Based Design

Zhihui Xiong<sup>1,2</sup>, Sikun Li<sup>2</sup>, Jihua Chen<sup>2</sup>, and Maojun Zhang<sup>1</sup>

<sup>1</sup>School of Information System and Management,  
National University of Defense Technology, 410073, Changsha, P.R. China  
xzhnudt@vip.sina.com, maojun@mail.iscas.cn

<sup>2</sup>School of Computer Science, National University of Defense Technology,  
410073 Changsha, P.R. China  
lisikun@263.net.cn, jhchen@nudt.edu.cn

**Abstract.** To facilitate the design of SoC (System-on-a-Chip), we present a hardware/software co-design environment called HSCDE. In this paper, some critical techniques related to HSCDE are revealed, including Platform-Based SoC modeling technology and ant algorithm based hardware/software partitioning technology. HSCDE environment divides SoC hardware/software co-design processes into three design levels, and it also supports the mappings among these design levels by two design mapping processes. Experimental results show that HSCDE effectively supports hierarchical Platform-Based SoC hardware/software co-design methodology, and further statistics reveal that an average of 10%~25% revisions on platform templates are required to get a new SoC design.

## 1 Introduction

SoC (System-on-a-Chip) integrates signal collection, signal conversion, data storage, signal processing, and input/output functionalities into a single chip. It has a number of advantages, including high speed, high integrity, low power, small size, and low cost. SoC has become a hot topic in VLSI design. However, design of SoC is very complex and SoC products have a short market window. Traditional "hardware first and software second" methods can no longer support SoC system design effectively. Two major means to solve these problems are hardware/software co-design and high level design reuse.

There are mainly Block-Based Design method and Platform-Based Design method for SoC hardware/software co-design [1]. The former emphasizes Intellectual Property reuse, constructing SoC system through integration of Intellectual Property cores. On the other hand, Platform-Based Design method is a co-design method introduced recently [2][3][4][5]. This method extends the concept of design reuse and pays more attention on high level reuse. Compared with Block-Based Design method, Platform-Based Design method not only shortens SoC development period, but also improves design reuse ratio and design quality. In fact, Platform-Based Design method is becoming a main stream method in SoC hardware/software co-design field.

A number of SoC hardware/software co-design environments have been reported [6]. Gupta developed VULCAN [7], the first tool for hardware/software co-design. It uses control/data flow graph to denote system models, and applies greedy method to implement hardware/software partitioning. Ernst's COSYMA [8] uses profile of system behavior model to guide hardware/software partitioning and synthesis. Ptolemy [9] is a well known heterogeneous hardware/software co-simulation system, which supports many system models (such as data flow, discrete events, finite state machine). Polis [10] system suits for SoC hardware/software co-design of real-time controlling applications. It uses an extended Finite State Machine (Co-design FSM) to model SoC system. SCE [11] is a successful hardware/software co-design system developed mainly by Gajski. It uses SpecC (an extended C programming language) to describe the system. SCE defines architecture refinement, communication refinement and hardware/software implementation refinement to achieve the mapping from SoC system model to Register Transfer Level SoC system. TIMA laboratory [12] in France and STARC center [13] in Japan are also constructing their hardware/software co-design systems.

Having analyzed all above SoC hardware/software co-design environments, we find that they share some common drawbacks: 1) These environments support Block-Based Design methodology and Intellectual Property reuse, but they ignore the importance of system reuse. This shortcoming makes them not suitable for Platform-Based SoC hardware/software co-design methodology. 2) These environments only support some phases of SoC design, but they cannot support all the phases of it. This shortcoming brings severe consistency problems during SoC design.

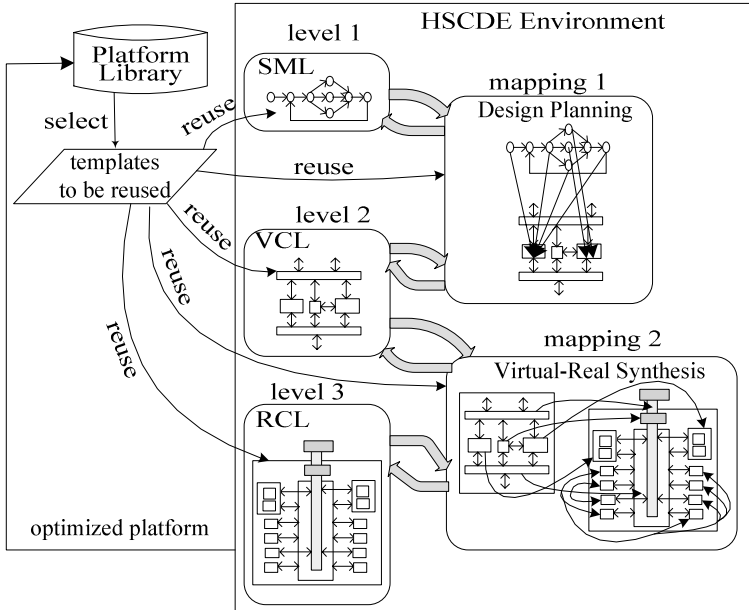
In order to solve these problems, we developed a new hardware/software co-design environment called HSCDE. In this environment, we have applied some critical and valuable hardware/software co-design techniques, including Platform-Based SoC system modeling and ant algorithm based hardware/software partitioning technology. HSCDE supports the hierarchical Platform-Based SoC hardware/software co-design methodology introduced in [14]. In HSCDE, three design levels defined, i.e. System Modeling Level (level 1), Virtual Components Level (level 2) and Real Components Level (level 3). Moreover, HSCDE provides two mapping processes for these levels by Design Planning (mapping 1) and Virtual-Real Synthesis (mapping 2).

We have done a simple case study and experimental designs for MP3 player, MPEG2 player and CDMA wireless communication SoC. Results show that HSCDE effectively supports the Platform-Based SoC hardware/software co-design methodology, and further statistics reveal that we need only an average of 10%~25% revisions on platform templates to get a new SoC design, so the average platform template reuse ratio is 75%~90%.

The next section introduces the overall structure of HSCDE. In Section 3, we present the three design levels in HSCDE in detail. Then, the two mapping processes in HSCDE (i.e. Design Planning and Virtual-Real Synthesis) are explained in Section 4. In Section 5, we describe a simple case study and some experimental results. Finally, we draw conclusions and discuss future work in Section 6.

## 2 Overall Structure of HSCDE

Fig. 1 shows the overall structure of HSCDE. In HSCDE, the SoC hardware/software co-design process is divided into System Modeling Level (SML; level 1), Virtual Components Level (VCL; level 2) and Real Components Level (RCL; level 3). Besides, HSCDE supports the two mapping processes by Design Planning (mapping 1) and Virtual-Real Synthesis (mapping 2), and it provides reuse templates via platform library.



**Fig. 1.** Overall structure of HSCDE

At the System Modeling Level, we use Constrained Taskflow Graph model [15] to describe SoC system functionality and performance constraints. Virtual Components Level is a "virtual design" level, and it abstracts the architecture of Real Components Level. Real Components Level is a "real design" level. This level includes detailed Register Transfer Level hardware design information and embedded software modules.

Design Planning performs the mapping from System Modeling Level to Virtual Components Level. It is used to do hardware/software partitioning and performance constraints assignment. On the other hand, Virtual-Real Synthesis maps Virtual Components Level SoC system to Real Components Level.

In HSCDE, three design levels ensure separation of behavior from structure and separation of computation from communication.

### 3 Design Levels in HSCDE

#### 3.1 System Modeling Level

System modeling is an important problem in SoC hardware/software co-design. In HSCDE, the System Modeling Level describes system behavior and performance with algorithms.

HSCDE supports Constrained Taskflow Graph model which is a type of variable granularity SoC description model. Constrained Taskflow Graph model describes task performance constraints by defining constraint attributes, and it describes task functionality by algorithm mapping. In this model, it defines subtask execution controlling machine to describe control structures, e.g. parallel, branch and loop.

Fig. 2 shows the modeling technology used in HSCDE, this technology supports model library management, model reuse and model customization. At the same time, this technology considers both reusability and flexibility of models.

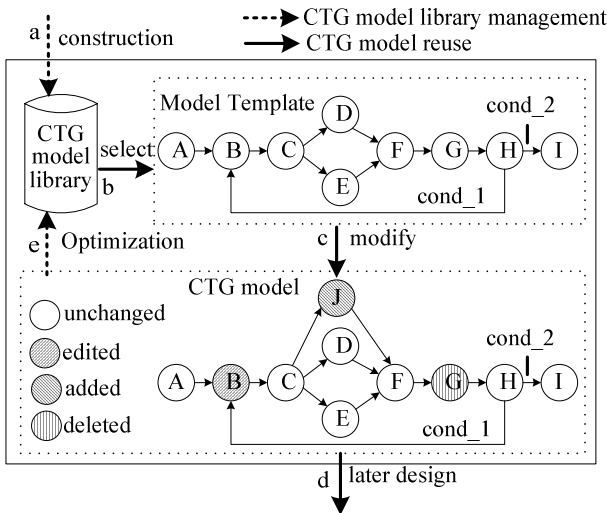


Fig. 2. Constrained Taskflow Graph modeling, reusing and modification

#### 3.2 Virtual Components Level

Virtual Components Level is the "virtual design" level in HSCDE. This level abstracts the SoC system architecture of Real Components Level. Use of Virtual Components Level makes it efficient to decrease the complexity of direct mapping from System Modeling Level to Real Components Level. As we can see in Fig. 3, Virtual Components Level includes Virtual Hardware Intellectual Property components (VHwIP), Virtual Software Intellectual Property components (VSwIP) and Virtual Communication Intellectual Property components (VCommuIP). The interconnection



structure between these components is shown in Fig. 4. Among them, VHwIP is the abstraction of some type of special purpose Intellectual Property cores (excluding micro-processor cores and DSP cores). It conceals the complex details of Intellectual Property cores. VSwIP is the algorithm procedure that will be mapped to embedded micro-processors (such as MPU, DSP). VCommuIP defines the interconnection and communication behavior of VHwIPs and HSwIPs.

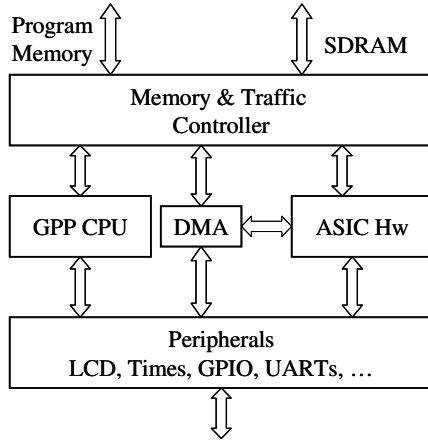


Fig. 3. Virtual Components Level SoC architecture

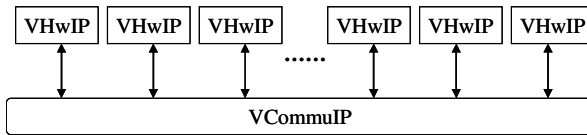


Fig. 4. Virtual Components Level SoC interconnection

### 3.3 Real Components Level

Real Components Level is the Register Transfer Level hardware/software SoC system. Fig. 5 shows the hardware centric view of Real Components Level SoC system platform. Real Components Level system can be divided into software part and hardware part. The software part includes real-time operating system, device driver APIs and application processes. The hardware part includes hardware accelerator modules (such as co-processor, DSP, ASIC) and input/output control devices.

Fig. 6 shows a detailed Real Components Level SoC hardware/software system platform used in HSCDE. From this figure, we can see that Real Components Level system platform includes the following information: 1) Structure and function description of application specific Intellectual Property cores. 2) Instruction set structure and resource configuration description of embedded processor cores. 3) Interconnection net description and embedded software module description.

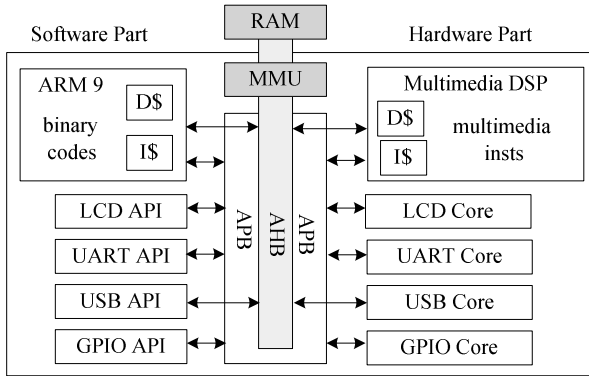


Fig. 5. Hardware centric view of Real Components Level SoC system platform

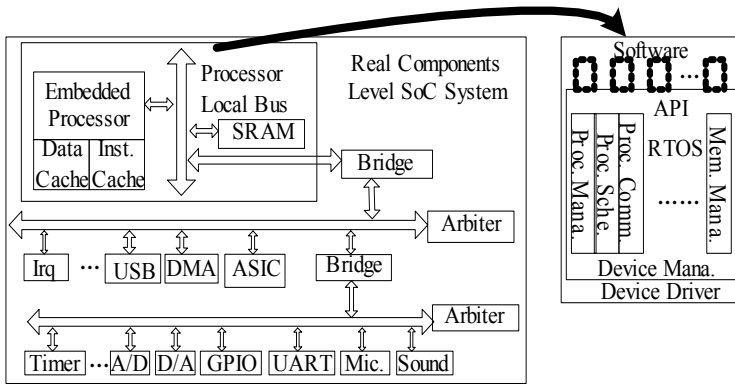


Fig. 6. Detailed Real Components Level SoC hardware/software system

## 4 Mappings Between Design Levels

There are two mapping processes in HSCDE, i.e., Design Planning and Virtual-Real Synthesis. The Design Planning process performs the mapping from System Modeling Level to Virtual Components Level. On the other hand, the Virtual-Real Synthesis process performs the mapping from Virtual Components Level to Real Components Level.

### 4.1 Design Planning

Design Planning performs the mapping of System Modeling Level to Virtual Components Level, there are two steps in this process, i.e. hardware/software partitioning and system task performance constraints assignment.

HSCDE provides automatic partitioning and interactive partitioning for SoC designers, and the automatic partitioning method is based on ant algorithm.

Dorigo introduced ant algorithm in [16]. In HSCDE, we make use of the system reuse feature of Platform-Based design method, and we develop a hardware/software partitioning technology that is based on ant algorithm. The basic ideas are: 1) Transform pre-verified partitioning results in the platform into initial pheromone, and when we run ant algorithm, we use these results to get initial pheromone. 2) Based on the initial pheromone generated, we run ant algorithm to find optimal partitioning results.

## 4.2 Hardware/Software Co-synthesis

The purpose of hardware/software Co-synthesis (Virtual-Real Synthesis) is synthesizing Virtual Components Level SoC system into Real Components Level. At this stage, VHwIPs will be synthesized to real Intellectual Property cores, VSwIPs will be synthesized to embedded software modules, and VCommIPs will be synthesized to real interconnection net.

The steps of Virtual-Real Synthesis are:

- (a) Virtual-Real components matching. In this step, VHwIPs will be attached to real Intellectual Property cores that satisfy the performance requirements, and VSwIPs will be attached to software processes that are scheduled by embedded operating system.
- (b) Interface synthesis of real components. According to the communication relationship described in VCommIPs, we get the real interconnection net through interface synthesis technology. During this step, it is possible to generate glue logics.
- (c) Compile and optimize embedded software modules. This step will transform and compile VSwIPs into object codes that can be run on selected embedded processor.

## 5 Case Study and Experiments

In order to show how users perform their hardware/software co-design tasks in HSCDE, we make a simple case study on the design of MP3 SoC. Then, we present experimental results on the design of MP3, MPEG2 and CDMA SoCs.

### 5.1 Case Study on the Design of MP3 SoC

Design of MP3 player is widely used as a case study in hardware/software co-design domain. So, we demonstrate the process of designing MP3 SoC in our HSCDE environment.

There are two major steps included: 1) Selecting or designing suitable platform to be reused. 2) Do some modifications on the selected platform, so as to achieve the

required functionalities and performance. Fig. 7 shows three design levels and two mappings while designing the MP3 SoC in our HSCDE environment.

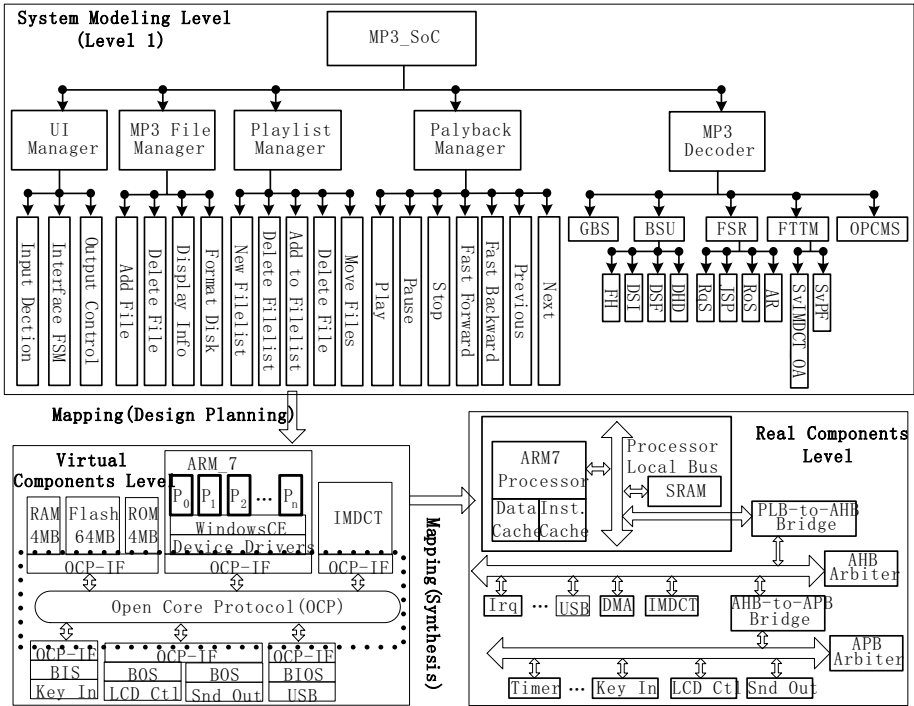


Fig. 7. Design levels and mappings during MP3 SoC design

### 5.2 Experimental Results

We have designed SoC systems for MP3 player, MPEG2 player and CDMA wireless communication decoder in HSCDE. Table 1 shows the results. In this table, the data in each grid means total count of modules and the number of modules that has been modified. For example, the data 15(8) in line 8 column 4 means that when we are designing the SoC for MPEG2 player. If we adopt the design methods that support traditional Platform-Based design environment, there are totally 15 software modules, and we need to modify 8 of them.

Results in Table 1 indicate that HSCDE supports Platform-Based SoC hardware/software co-design methodology well. This design environment overcomes the difficulty of direct mapping from SoC system model to Register Transfer Level. Statistics reveal that an average of 10%~25% revisions on platform templates is required to get a new SoC design. So, we can achieve platform template reuse ratio by 75%~90%.

**Table 1.** Comparison of different SoC design using different design methods

		Co-design Environment Type		
		Block-Based Co-design Environment	Traditional Platform-Based Co-design Environment	Hierarchical Platform-Based Co-design Environment
MP3	System Models	19(19)	19(4)	19(4)
	Hardware Modules	n/a	16(5)	9(2)
				16(3)
Software Modules	n/a	11(7)	7(2)	
				11(4)
MPEG2	System Models	28 (28)	28 (7)	28 (7)
	Hardware Modules	n/a	21(9)	9(3)
				21(5)
Software Modules	n/a	15(8)	10(4)	
				15(6)
CDMA	System Models	35 (35)	35 (6)	35 (6)
	Hardware Modules	n/a	19(7)	7(2)
				19(3)
Software Modules	n/a	24(9)	8(2)	
				11(4)

## 6 Conclusions and Future Work

This paper introduces a novel SoC hardware/software co-design environment and describes some critical techniques in this environment, such as Platform-Based SoC system modeling technology and ant algorithm based hardware/software partitioning technology.

This environment supports hierarchical Platform-Based SoC hardware/software co-design methodology. Experimental results prove that the proposed SoC design environment is suitable for SoC system design.

Future work on the proposed HSCDE environment include: 1) Since product cost and performance are becoming more and more important, we will implement new features with regard to cost and performance in HSCDE. 2) For this moment, HSCDE does not put more effort on low power design, so we will also add power evaluation and low power design methods in HSCDE.

## Acknowledgements

The work presented in this paper is supported by the National Natural Science Foundation of China (No. 90207019) and the National 863 Program of China (No. 2002AA1Z1480).

## References

1. Chang, H., Cooke, L., Hunt, M., Martin, G., McNelly, A., Todd, L.: *Surviving the SoC Revolution: A Guide to Platform-Based Design*. Kluwer Academic Publishers (1999)
2. Keutzer, K., Newton, R., Rabaey, J., Sangiovanni-Vincentelli, A.: System-level design: Orthogonalization of Concerns and Platform-Based Design. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems* 19 (2000) 1523-1543
3. Carloni, L.P., Bernardinis, F.D., Sangiovanni-Vincentelli, A., Sgroi, M.: The Art and Science of Integrated Systems Design. *Proceedings of the 28th European Solid-State Circuits Conf. (2002)* 25-36
4. Jiang Xu, Wayne Wolf: Platform-Based Design and the First Generation Dilemma. *Proceedings of the 9th IEEE/DATC Electronic Design Processes Workshop (2002)* 21-23
5. Bob Altizer: *Platform-Based Design: The Next Reuse Frontier*. Embedded Systems Conference, San Francisco (2002)
6. Zhang, L.F.: *Research on Techniques of Hardware/Software Co-synthesis and Virtual Microprocessor*. PhD thesis, National University of Defense Technology, Changsha, P.R. China (2002)
7. Gupta, R.K.: *Co-synthesis of Hardware and Software for Digital Embedded Systems*. PhD thesis, Stanford University (1993)
8. Ernst, R., Henkel, J., Benner, T., Ye, W., Holtmann, U., Herrmann, D., Trawny, M.: The COSYMA Environment for Hardware/Software Cosynthesis of Small Embedded Systems. *Microprocessors and Microsystems* 20(1996) 159-166
9. Davis, J.: *Ptolemy II - Heterogeneous Concurrent Modeling and Design in JAVA*. University of California at Berkeley (2000)
10. Balarin, F., Giusto, P., Jurecska, A., Passerone, C., Sentovich, E., Tabbara, B., Chiodo, M., Hsieh, H., Lavagno, L., Sangiovanni-Vincentelli, A., Suzuki, K.: *Hardware-Software Co-Design of Embedded Systems: The POLIS Approach*. Kluwer Academic Publishers (1997)
11. Abdi, S., Shin, D., Gajski, D.D.: Automatic Communication Refinement for System Level Design. *Proceedings of ACM IEEE Design Automation Conference (2003)* 300-305
12. TIMA Laboratory at: <http://tima.imag.fr/sls/research.html>
13. STARC, Project VCDS Development at: [http://www.starc.jp/kaihatu/vcdsgr/vcds\\_intro\\_e/4nofrm.html](http://www.starc.jp/kaihatu/vcdsgr/vcds_intro_e/4nofrm.html)
14. Xiong, Z.H., Li, S.K., Chen J.H., Wang, H.L., Bian, J.N.: Hierarchical Platform-Based SoC System Design Method. *Acta Electronica Sinica*, 32 (2004) 1815-1819
15. Xiong, Z.H., Li, S.K., Zhang, L.F., Chen, J.H.: A New SoC System Modeling Method. *Proceedings of the 8th Int'l Conf. on CAD/Graphics (2003)* 157-161
16. Dorigo, M., Maniezzo, V., Colomi, A.: Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man and Cybernetics, Part-B*, 26 (1996) 29-41

# A Computer Supported Collaborative Dynamic Measurement System

Peng Gong<sup>1</sup>, Dongping Shi<sup>1</sup>, Hui Li<sup>1</sup>, Hai Cao<sup>1</sup>, and Zongkai Lin<sup>2</sup>

<sup>1</sup>Deptment of Computer Science,  
Shandong University at Weihai, Weihai, P.R. China  
gongpeng@sdu.edu.cn

<sup>2</sup>Institute of Computing Technology,  
Chinese Academy of Sciences, Beijing, P.R. China  
Lzk@ict.ac.cn

**Abstract.** This paper presents our recent research work on collaborative dynamic measurement. The main purpose of this research is to design the structure, function and working model of a collaborative dynamic measurement system. This paper first introduces the concept of Computer Supported Collaborative Dynamic Measurement System (CSCDMS) which includes the working model of collaborative measurement group, CSCWMS structure, and collaborative dynamic measurement data processing. The primary objective of CSCDMS is to share the distributed data resources, to reduce the cost and increase the measuring accuracy under the working model of dynamic measurement data processing. The systematic analysis shows that the working efficiency of Collaborative Dynamic Measurement System is much higher than that of the isolated dynamic measurement system.

## 1 Introduction

Dynamic measurement is the primary technique of measuring techniques currently. Compared with traditional static measurement, it owns dynamic, random, asynchronous, correlated and time sequential characteristics. So the method of measurement and data processing changes dramatically. As a result, dynamic measurement cannot use the theory of the traditional static measurement. At the same time, the modern manufacturing industry requires higher accuracy and precision. The production of better equipment to improve the accuracy proves to be costly and infeasible with the respect to technique. The research of dynamic measurement can offer a good method to increase the measuring accuracy and decrease the measuring cost.

At present, the research and development on dynamic measurement data processing still stays on the stage of single computer controlled measurement system. The single computer controlled system offers a simple person-to-computer exchanging environment. The working procedure of such system is usually as follows: (1) Collecting the measuring data from the outside measuring equipment. (2) Selecting a signal processing model to set up a simulation model. (3) Modifying the simulation data. (4) Modifying the simulation model. (5) Assessing the model accuracy. This kind

of model can only support the work of single measuring engineer, which can only offer one simulation model for data processing and accuracy assessment. So the person-to-computer exchanging model is concluded as a time consuming, weak data processing model.

Recently, Chongqing University of China has developed a long distance data processing method based on Matlab Web server technique, which realizes the long distance data collecting, transferring, storage and processing, and data sharing via server database. But the working model keeps the traditional person-to-computer exchanging method which is impossible to improve the system measuring efficiency. In order to develop a new dynamic measurement system which offers the functions such as distributed working, fast data processing, communication between measuring members, and the platform for collaborative measuring work, computer Supported Collaborative Dynamic Measurement System (CSCDMS) can be a good solution. CSCDMS integrates the techniques of control, network, CSCW and signal processing. CSCDMS is capable of realizing the function of Group Management (GM), Public Service (PS), Data Management (DM) and Control Management (CM).

## **2 Group Working Model of Collaborative Dynamic Measurement**

CSCDMS offers a group measuring platform for collaborative work, which changes the model of person-to-computer exchange and realizes the exchange between measuring members. The data collected from the measuring equipment is stored in the initial database and then is submitted to the measuring member by the initial database for modeling, computing, analysis and assessment. That type of collaborative working model can make measuring members fulfill the task in different places and assess the measurement. So the research of dynamic measurement technique is necessary to modern industry.

The working procedure of collaborative dynamic measuring group includes following steps:

- (1) Under systematic collaborative management, collaborative measuring members set up a working group via collaborative tools (group white board).
- (2) A certain measuring member applies for measuring data towards the system. After receiving the application, the system will give orders to drive the measuring equipment to do measuring work and then send back the signals for storage and to group members.
- (3) Processing the measurement data. The member management of the system supports the single working of every measuring member. The working of the member includes selecting a signal processing technique from the tool base and setting up a simulation model and then processing the simulation data and assessing the simulation model. During this step, the system offers a model of person-to-computer exchange.
- (4) After all the members of the measuring group finish their required processing, the results will be shared among members of the group through collaborative tools (data processing white board). The measuring members can also discuss the data processing techniques and assess the selected method via collaborative tools.



The key advantage of the designed person-to-person exchange model is that, since multiple measuring members process the data and compare the models, the best model can be selected. The person-to-person exchanging method shares the measurement system resources, improves the communication among measuring members, offers a platform for measuring members to do research work on Modern Error Theory, and increases the measuring quality and efficiency.

### 3 The Structure and Function of CSCDMS

CSCDMS aims at processing dynamic measurement data collaboratively. With the respect to the structure level, CSCDMS consists of the sensor system and computer collaborative processing system. In order to process measuring signals and realize timing control, we add the functional block of the control management block into the system. The proposed model is shown in Figure 1.

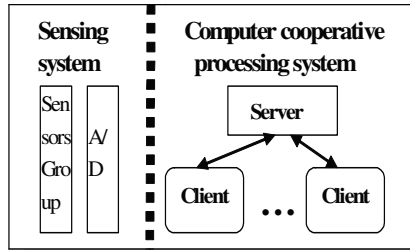


Fig. 1. Model of the system

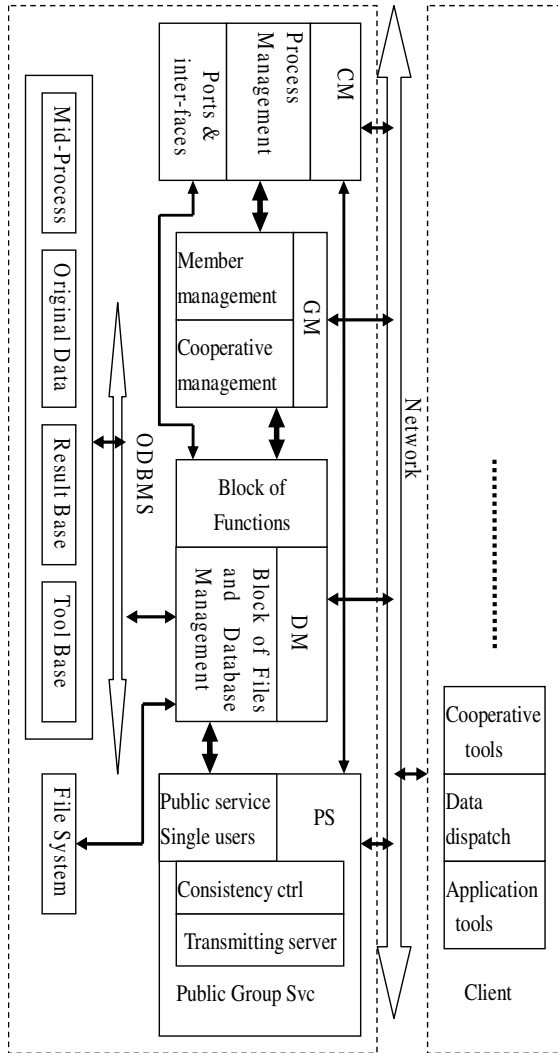
The realization of CSCDMS technique includes the following issues:

- (1) CSCDMS can collaboratively finish the function of data collecting, transferring and storage, which can offer measuring data to measuring members precisely in real time. The application of Java multi-processes mechanism can recognize different types of signals, which is the main method of collaborative multi-signal sources collection technique.
- (2) The GM and PS of CSCDMS offer services for group data processing and collaborative work, which use the functions of consistency control and transferring service to ensure the running of the group collaborative work. Then, the system realizes the person-to-person exchange and asynchronous measurement.
- (3) CPU can assign orders to different tasks and processes. We can make use of Java technique and its multi-processes to share the distributed database. The dispatch of data is finished by the interface between users and the data-managing block of servers. The interface is in charge of sending the data to users and applying for various data, including raw data and mid-process data.

#### 3.1 Structure of CSCDMS

CSCDMS adopts the client-server architecture, by setting up functional models in the server, which include Group Manager (GM), Public Server (PS), Data Manager (DM),

and Control Manager (CM). GM manages the members of the group. PS works for the data processing and cooperation of the group. DM controls various stoppages and upholds system working. CM's main responsibility is to monitor all the information while the system is functioning, in another word to fulfill the monitoring task. We can see the whole procedure in detail in the structure as shown in Figure 2.



**Fig. 2.** Structure of the system

Java can be used to implement CSCDMS. It can process TCP/IP communication by programming on Java lab and then make use of URL address to visit other objects

conveniently. It can finish the net links on all kinds of levels through java.net package. The socket type can provide reliable mobile net link, and create distributed client and server applications.

### 3.2 The Function of CSCDMS

Client/Server model can be defined as a special type of collaborative processing model. The whole application program exists both in the client terminal and the server. Not only the server but also the client terminals participate in the application program processing. In that model the software is used to finish a certain function of the application program and the hardware resources can offer the necessary support of the software cooperation. So, Client/Server model includes not only the collaboration within the software but also the exchanging interaction within the hardware.

System service can take the responsibility for data receiving and saving. In the system, the server takes the responsibility for the data receiving and storage and manages the members working at the client terminals so as to sustain the normal functioning of the system and to give collaborative services to the members. The clients mean the different computers that are linked via the network to the server. The users of the system need not take the position of the terminal into consideration. What they need to do is only registering the clients, selecting a data processing model, setting up the group and group discussion. The client terminal can provide the users with the application tools to complete the function of data transmitting and collaborative work.

The realization of application tools includes the function of member registration, the function of data processing, the using of the tool base for selecting algorithm, the function of the result storage and so on.

The function of data dispatch is the interface between the client terminal and the data-managing block of the server. It takes the responsibility for various data registration of members and sending the data needed to the client terminals dynamically, which include not only the raw data, but also the data collected during the collaborative measuring work.

Collaborative tools can mainly fulfill the tasks such as member registration, group members' cooperation and asynchronous platform of the announcement function. The designing principle of collaborative tools is to express a large amount of information precisely with relatively simple interfaces, especially to offer the necessary support to the group cooperation.

Through GM, a client can be entitled to dispatch the data. If that data are under discussion or storage at the same time by a certain group, then GM will add the client to that group. GM can also set up single client application while waiting for new client to join the collaboration and make up a new group. When the group discussion is over, GM should cancel the group.

The single service of PS processes the function of the person-to-computer exchanging dynamic measurement. Group collaborative service can ensure the cooperation among different clients. They are the key sectors of the CSCW realization. They guarantee the running of the group cooperation, at the same time offer consistency control and redistributing service.

The interface control of CM is mainly used to receive the raw data sent by sensors. Process management works according to the working flow and stores the relevant data

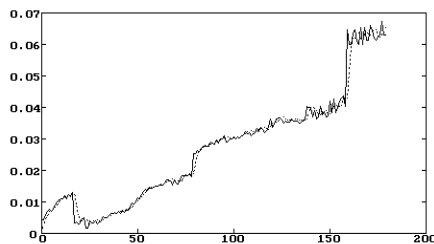
into the special process database during the whole period of system running. When there are false operations or incremental requirements, the data can be obtained and restored by searching the process database. As for every group and member, the system can evaluate their working efficiency according to the historical records in the process database. Those functions of the process management benefits both system itself and the commercial activities. Those functions are further extension based on dynamic measurement and collaborative mechanism.

DM includes the raw database storing raw data linked by ODBC. PM can be realized by creating process database and result database. File system and Tool base cannot be linked by ODBC where they need the direct management. File system has the effectiveness of the multimedia data access. Tool Base is a group of data processing algorithms designed according to the trend of sharing methods. After a client's choosing a certain algorithm, the system will dispatch the algorithm automatically and install it to the client's computer for using under the Java technique support. At the same time, the Tool base will update continuously to adapt to the need of real-time data processing, which means client will have more choices in algorithms. DM also sets up several extra interfaces or ports in order to adapt to the access of different data types.

## 4 Conclusion

The modeling mechanism of modern signal processing methods is different, so the results of the simulation modeling application are different. It is known that if we apply different simulation models to the same group of data, the results of different models can be different. Figure 3 and Figure 4 show the curves of processing the same group of data in person-to-computer exchanging single system via BP Weight Studying Neural Network model and Combination Model of Grey System Theory respectively. Both 3 and 4 are data processing models. The Y axis indicates  $x(t)$ , the input signal of dynamic measurement data processing which is produced by the combination of measured real-time value and measuring equipment. The X axis indicates  $t$ , the time.

Figure 3 shows the results produced by weight studying BP neural network and Figure 4 shows the results produced by Grey system theory model.



**Fig. 3.** Result processed by BP neural network (Unit: mm)

In Figure 3, the bold line indicates the results of error measurement, the light one shows the results obtained after the data processing using BP Weight Studying Neural Network model. Due to the strong character owned by the Neural Network model and

the distributing information weight value varying in the whole net, some unit errors will not influence the whole information processing function of the net. The Neural Network model has the inaccuracy containing character, which means the abnormal values obtained after the dynamic measurement will have little influence on the whole data processing and get more accurate processing results.

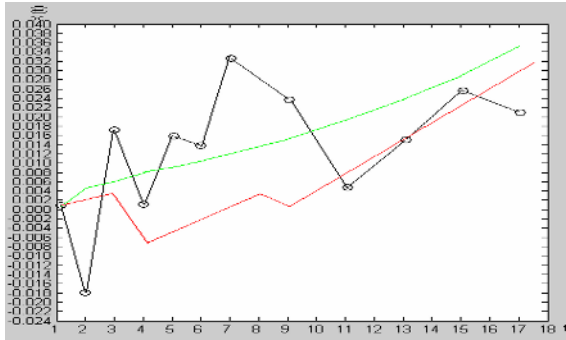


Fig. 4. Result processed by Combination of Gray system theory (Unit: mm)

Figure 4 shows the results of applying the Combination Model of Grey System Theory. Grey Theory Model is based on the smooth discrete functions, which focuses on describing the trend of system certainty and owns a strong controlling capability. But the Grey Theory Model can only weaken the disturbance of random factors. In order to inflect the whole dynamic measurement process, the model has to complete the data processing combining with the model of describing random factors. The Grey Theory Model has a disadvantage of bad errors containing ability, any disturbance from outside can cause the false of the model.

We set up two simulation models towards a group of measuring data. That is called as one group of data towards multi-models' data processing method. The best real-time dynamic measuring data processing method is to set up different types of models, and assess the measuring results of them for selecting the most accurate model in the shortest period. Generally speaking, the quantity of computing is less, the real-time measurement is more efficient. For example, there are a group of data and m types of simulation models, the system needs  $1 * m$  runs of computing to finish the task. If there are n sets of data groups and m types of simulation models, there will be  $m * n$  runs of computing. Take the assessment of the optimal model curve into consideration, the precision degree of  $1 / (m * n)$  is higher than that of  $1 / m$  model. Via the upper analysis, it can be concluded that the measuring time increasing, the computing runs will increase. As a result, the computing accuracy will also be improved. It is proved that the high accurate measurement needs the long period of time. The key problem to be solved is how to gain the high accuracy measuring results within the shortest time. In the single computer supported system, every measuring member completes the whole process of assessment and measurement. The system needs  $1 * m$  runs or  $n * m$  runs of computing to finish the work. If there are x clients, the system has to allocate  $x * 1 * m$  or  $x * n * m$  runs of time to clients. This model of serial processing cannot realize the effect of optimal dynamic measuring real-time data processing. CSCDMS is a model of parallel

processing, which supports the members to research and finish the dynamic measuring data processing. Under the model of one group of towards several processing methods,  $x$  measuring members can processing  $m$  kind of different models, ignoring the influence of other factors, the system only allocates  $1/m$  time period to satisfy the time requirement for each measuring member. By the analogy, we can see that under the multi-data-groups to multi-models condition, if there are  $n$  groups of data and  $m$  kinds of models, the  $n/m$  period of time will meet the dynamic data processing need.

The paper designs the structure of CSCDMS, and analyzes the parallel working model of CSCDMS logically. CSCDMS can not only increases the working efficiency of measuring, but also select the best method from the models. In order to realize the platform of CSCDMS, the problems such as simulation models classification, recognition, storage, the interface processing, the integral constraint of Web transferring data, and security, are to be solved. All in all, CSCDMS extends the theory and technique of dynamic measurement, explores a new method for dynamic measurement.

## References

1. Gong, P.: Research on the Theory and Application Technology of Grey Model for Error Correction of the Dynamic Measurement, PhD Thesis, the School of Instrumentation, Hefei University of Technology. (1999)
2. Xu, B.: Research on the Application Collaborative Technology, PhD Thesis, the School of Instrumentation, Hefei University. (2000)
3. Wigand, R., Picot, A., and Reichwald, R.: Information, Organization and Management. Singapore: John Wiley & Sons, (1997)
4. Baresi, L., Casati, F., Castano, S., Fugini, M.G., and Mirbel, I.: Wide Workflow Development Methodology. Proc. ACM WACC (Work Activities Coordination and Collaboration) 99, San Francisco, CA, USA (1999)
5. <http://www.rational.com/media/uml/post.pdf>
6. Biran, A. and Breiner, M: MATLAB 6 for Engineers. Prentice Hall, New York, (2002)
7. Li, W., Gong, W., Qin, L., Liu, J.: A new method to develop the data processing system based-on the Matlab and Web technology, Proceedings of the Eighth National error theory and application conference (year2004).

# A Collaborative Management and Training Model for Smart Switching System

Xiaoping Liao<sup>1,2</sup>, Xinfang Zhang<sup>1</sup>, and Jian Miao<sup>2</sup>

<sup>1</sup> CAD Center, Huazhong University of Science & Technology,  
Wuhan, China, 430074  
zxfang@hust.edu.cn

<sup>2</sup> School of Mechanical Engineering, Guangxi University,  
Nanning, China, 530004  
xpfeng@gxu.edu.cn

**Abstract.** In order to reduce the training cost and improve the efficiency of using the Smart Switching System (SSS) of electric power substations, this paper proposes a collaborative management and training model for Smart Switching System using the ASP (application service provider) mode. The proposed model facilitates the communication between system user companies and the system provider. It allows a user company to evaluate the Smart Switching System and train its engineers over the Internet during the process of purchasing and deployment of the system.

## 1 Introduction

In a collaborative design process where multiple designers work together a complex design project, these designers share data to solve the design problems that cannot be solved by any single designer. The process of managing the data possessed by both designers and computers has an important impact on collaborative design practice [1,2].

Generally, the Smart Switching System (SSS) of electric power substation should achieve follow goals [3]: Firstly, the system operation mode will be graphics-oriented, using single-line diagram and protection facility diagram customized by users. Secondly, operators can visually select the specific component to produce a message event. Thirdly, the built-in interlocking rules will be used to check whether the switching sequence violates any interlocking constraints. In conclusion, the component's state will be changed and the switch sequence will be generated at the meantime, or warning message of misoperation will be displayed.

However, during the process of purchasing and deployment of SSS, user companies need to be trained by a system provider. Learning and designing specified construction of SSS with line diagram and specifications, the user companies require the system provider to provide adequate supports to solve some technical problems during the design and training process. Therefore, in order to reduce training and consulting costs, it is necessary and useful to develop a collaborative management platform using ASP (application service provider) mode, which can share resources. Therefore,

We propose a collaborative management and training model based on ASP mode to provide an efficient and effective support to the collaborative design and training of the Smart Switching System. The proposed model mainly adopts the method of on-line interaction between the system provider and user companies to improve the efficiency of using SSS.

## 2 Collaborative Management and Training Model

This section introduces the proposed collaborative management and training model based on the ASP mode. It includes the architecture of proposed model, collaborative management structure and implementing techniques.

### 2.1 Architecture of Collaborative Management and Training Model

Figure 1 shows the proposed architecture of collaborative management and training model based on the ASP mode. The model consists of application support model, running process logic layer and shared software application layer. The application support model offers a series of supporting mechanisms for the collaborative management and training process. According to detailed collaborative management and train function requirements, the support model is divided into several sub-models. These sub-models include design parameter definition model, rule/case knowledge and representation model, and interlocking constraint model.

The working process of running process logic layer can be described as follow: During the process of choosing a SSS, the user companies require to estimate SSS's function and train their designers and operators, the system provider provides a supporting ASP platform using the collaborative design and training model. To design specified SSS, a user company submits a requirement document with product drawings and specifications to the platform, and interacts with the system provider to obtain adequate information at the same time. After the system provider receives the product drawings and specifications, a network meeting will be held on the platform among several distributed working groups belonging to system provider and the user company. Detailed procedure is described as follow: (1) if some technical problems (or other identified problems) cannot be solved by the user company, the development and design groups of the system provider may provide an on-line technical support through the platform; (2) if some technical problems (or other identified problems) cannot be understood, the training groups of the system provider may provide an on-line training support through the platform; (3) the SSS design schedule is confirmed, and, during the communication process between the system provider and the user company, design errors in drawings can be checked out and fixed on-line.

There are many kinds of cooperation patterns among the user companies and the system provider, including substation line diagram design, parameter definition and edition, and switching operation and solutions to various problems. The proposed collaborative model has obvious advantages that the system provider and user companies can build uniform information system and collaborative architecture, implement information sharing, and realize resource optimization.



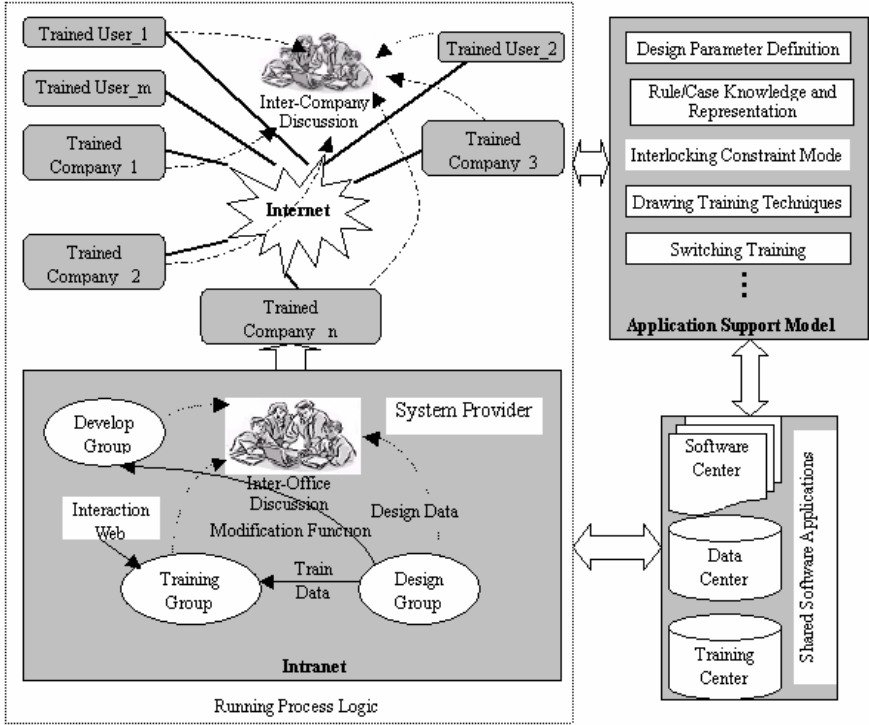


Fig. 1. Architecture of collaborative management & training model

**2.2 ASP Mode for Collaborative Management and Training**

An Application Service Provider (ASP) is a company using central computer systems (hosts, Unix systems, etc.) for the purpose of renting software to its users [4]. It is one type of professional service agency to provide application services or business solutions, which includes configuration, rental, administration, and other technology supports, and helps companies or individuals to acquire application services quickly and easily. The ASP gives its users remote access to software applications instead of downloading these applications. The users actually login the ASP’s computer system and run the applications at the provider’s server side, with only the results being downloaded.

There are significant advantages by adopting the ASP mode. For example, it is an ideal way for smaller companies which do not frequently use software applications and cannot burden expensive purchasing costs of these applications. Also, with an ASP, there is no need to purchase special applications for different operating systems. An application that only runs on a Unix system, for example, can be used from distributed clients with any operating systems. Another advantage is that some application software containing private algorithms can be rewritten so that it can only be run on an ASP platform. Competitors can only use the software but not download it to

their own systems or modify it. Therefore, the intellectual property of the software company is fully protected.

The use of an ASP help to meet company's increasing service demands continually. ASP is able to provide the latest business applications with less time for implementation. An ASP may provide their application services to other companies or users. For example, after Smart Switching System is build into a shared software application (SSA) using the ASP mode that is managed by the system provider, the user companies may login the SSA to evaluate SSS and use it, then make the decision for purchasing They may also take part in the training and designing activities.

The architecture of the ASP mode has been standardized, which consists of basic resource layer, system control layer, application service layer, and system entry.

In Figure 1, the resource layer is the foundation for storing shared resources information and basic database in data center. System control layer can monitor collaborative management and training resources, connect the upper application and lower resources like an adapter, and synchronize with related information to achieve data sharing between the system provider and user companies which attend to solve some business and technology problems. Application layer can provide tools to solve problems such as the shared application for SSS. System entry will enable enterprises or individuals to access all the tools and functions provided by the system. The application interface enables the data and work to transfer among design, process systems of allied system provider and companies and individuals. Another important tool of system entry is user collaborative management for user registration.

### 2.3 The Collaborative Management Structure

The platform server manages all training information, which is built on the SQL database. Its data structure is defined as follows:

*User Account:* includes User Name and User Password.

*User Basic Information:* such as Organization Name, Type and Relationship etc.

*Specified Diagram File Path:* to directly edit/modify it and be stored in the server.

*Training Subject:* the problem provided by users.

*Submitted Time:* time when the problem is generated, and time when the problem is solved.

*Subject State:* true or false or a predicate, showing whether a problem is solved.

*Problem Comments:* explanation for generating the problem and solving the problem.

In order to manage all training information effectively, the supported company should develop the interactive Web pages containing the above data structure and specifications written by XML on the World Wide Web [5].

In general, the system architecture consists of a four-tier structure: a thin client, a Web server, a business application server, and a database. All business logic locates on business application server, presentation and dialog on the client, and data services on the server. Therefore, all training information may be developed and built based on the business application server.

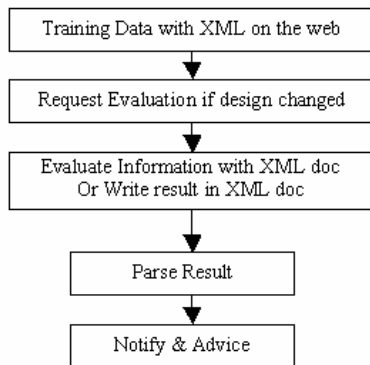
### 2.4 Developed Approaches of Collaborative Management and Training Model

The primary objective of implementing the collaborative management and training mode is to provide a system and framework for managing and training’s integration and customization, which help users to make use of knowledge supplied by other users (or designers). This knowledge sharing is the foundation of implementing the training model.

The user’s activities in the application mainly include working in the Internet/intranet, intercommunicating with users, and supporting company collaboration. Being easy to manage collaboratively, the proposed system set three types of users:

- *Normal user*: This is an information consultant, i.e., a curious/training person who wants to actualize his/her information about creating a specified system.
- *Expert user*: This is a system worker, i.e., a person who generates knowledge for the system in such a way that normal users are able to look it up. Any expert user may combine his/her own contribution with that of other experts’ or other normal users on the same topic, then, give the proper feedback information.
- *Administrator*: This is the person in charge of the system to keep it working correctly. Another responsibility of the administrator is the management of both the users and the topics that the system is to deal with.

Under the proposed system environment, a user can carry out the design while browsing an interactive Web page containing the design specifications written in XML on the World Wide Web. The client module connected to the design application writes these data in XML document form for Web publishing. An evaluation is submitted to the evaluation server. The evaluation server reads and parses all information from various XML documents on the Web. Evaluated results are written in XML document form and published on the Web. Then, the client module parses the result and takes related actions, e.g., notifying the *Expert user*, getting a help, giving some advices, and so on. Figure 2 shows the collaborative training processes.



**Fig. 2** Training result evaluating process

### 3 Implementation of Proposed Collaborative Model

#### 3.1 The Explanation of Smart Switching System

Based on the component technology, a Windows-based graphical user design interface (GUDI) has been developed [6], which assists and guides users to interact with the SSS, prompts them to find important information, setups the parameters of symbol objects, customizes the substation configuration and the correlative knowledge base for a particular Smart Switching System.

The simulation process begins with the drawing of the single-line diagram of the system. To draw a component on the screen, a user will select the component from the menu bar or tool bar. Then the component is drawn on the screen as the mouse left button is clicked a right position and the left button pressed is dragged to a desired rectangle area where the size of the selected component is displayed. At the meantime, the parameters of object-oriented component may be edited and modified; the large component based on a distribution feeder bay is auto-linked to implement aligning by various subcomponents. Also, the user can group a set of selected components to copy/cut or group move and can paste them on the screen. The single-line diagram of the Yulin Substation, drawn on the GUDI's main windows, is shown in Figure 3.

Once the substation single-line diagram has been completed in the GUDI, the rule-based and case-based expert decision system will be generated. Then, the user may execute the graphical user switching interface (GUSI), which is developed according to the particular task, the substation configuration and the operation arrangement, to simulate a switching event in the system or run process test, and generate the switching sequences and visualize outputs.

The SSS developed by the authors includes the following major modules:

- *Computer Aided Draft (CAD) Model*: A user may draw a single-line diagram, facility protection diagram and switching sequence table of substation.
- *Geometrical feature constraints*: to meet graphics position-topology and intelligent connecting with components.
- *The interlocking constraints of switching action amongst components*: to check whether the switching sequence violates any operating regulations described by case/rule-based reasoning system (CBR/RBR), in other words, to implement decision-making process.
- *The constraint predicates for knowledge representation of the components*: the knowledge of switching sequences and warning message is expressed by the attached predicate calculus.
- *Users design and simulation process*: the design and simulation process implies that in addition to the basic design—all external factors of possible relevance to the geometric links amongst components, knowledge representation of switching sequences and decision-making process of case/rule-based reasoning (CBR/RBR) are being considered in all stages of the design progress.

With regard to the collaborative management and training model of SSS, it includes the above training guidelines among the users, such as Computer Aided Draft (CAD), geometrical feature constraints, the interlocking constraints of switching action amongst components, etc.

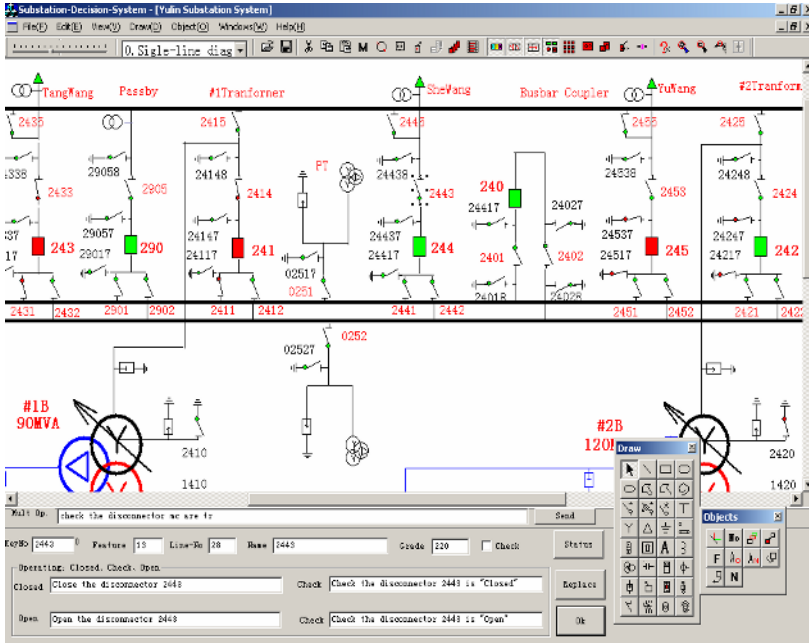


Fig. 3. Graphical user aided interface for the SSS

### 3.2 The Implementation of Collaborative Management & Training Model

In the proposed environment, a user can carry out the design while browsing an interactive Web page for training information as shown in Figure 4, which contains the design specifications written in XML on the World Wide Web.

The Web page as shown in Figure 4 can implement the collaborative management and training function of Smart Switching System, its functions and training schedules include:

- Create a user account;
- User access;
- User Management: stores all user information and training information.
- Design Parameter Definition: provides explanation of parameter definition and correlative examples.
- Rule Knowledge: provides explanation and correlative examples about rule knowledge.
- Case Knowledge: provides explanation and correlative examples about case knowledge.
- Distribution Feeder Bay: provides explanation and definition about position-topology and intelligent connecting with components.
- Interlocking Constraint: exhibits all correlative operating regulations.
- Knowledge Representation: explains the knowledge representation of switching sequences and decision-making process of case/rule-based reasoning (CBR/RBR).


- Computer Aided Draft Training: exhibits and trains how to draw a single-line diagram, facility protection diagram and switching sequence table of substation.
- Computer Aided Switching Training: exhibits and trains how to operate and generate the switching sequences and visualize outputs.

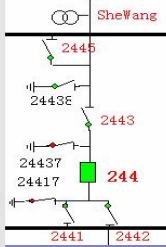
**Substation Smart Switching System**

User Id: ChangWang  
 Password:

---

**Cooperative Management Schedule**

Design Parameter Definition:  
 Rule Knowledge:  
 Case Knowledge:  
**Distribution Feeder Bay:**   
 Computer Aided Draft Training:  
 Interlocking Constraint:  
 Knowledge Representation:  
 Computer Aided Switching Training:  
 Resign Training(GUDI):  
 Operating Training(GUSI):



Bay NO:   
 Substation:   
 Subject:   
 Key word:   
 Advices:

**Fig. 4.** Interactive Web page for training

## Message Window

List	Problem	Write time	User	Status
1	Design Parameter Definition	2003-5-23 8:30	58592587	<input type="checkbox"/>
2	Rule Knowledge	2003-5-25 7:55	45785231	<input type="checkbox"/>
3	Case Knowledge	2003-8-10 13:50	44824343	<input type="checkbox"/>
4	Distribution Feeder Bay	2003-8-15 14:40	58592587	<input type="checkbox"/>
5	Computer Aided Draft Training	2003-8-25 7:14	58592587	<input type="checkbox"/>
6	Interlocking Constraint	2003-10-15 20:45	58592587	<input type="checkbox"/>
7	Knowledge Representation	2003-10-20 9:23	58592587	<input type="checkbox"/>
8	Computer Aided Switching Training	2003-11-10 10:15	58592587	<input type="checkbox"/>
9	Resign Training (GUDI)	2003-11-25 8:20	58592587	<input type="checkbox"/>
10	Operating Training (GUSI)	2003-12-2 19:14	58592587	<input type="checkbox"/>
11	Interlocking Constraint	2003-12-15 11:15	58592587	<input type="checkbox"/>
12	Distribution Feeder Bay	2003-12-30 9:45	58592587	<input type="checkbox"/>

**Fig. 5.** Message window of collaborative management

For a client, when user ID and password are correctly entered, he/she may get into the client's Web page to define a problem, then submit it to the server of software Support Company.

At the server side, when Message Window (as shown in Figure 5) receives a client's message, the receiver points to this message and opens the server's Web page that is the same as the client's as shown in Figure 4, then the corresponding department of the system provider will provide a solution to this problem, or directly go into the shared software application to modify the specified problem, finally inform the user of detailed modifications.

For example, when a client edits a problem of "Distribution Feed Bay" in collaborative management schedule Web page, which includes all training information, after submitting the problem information, the Message Window as shown in Figure 5 will receive a record information, such as user code: 58592587, Received Time: 2003-08-15 14:40, State: False (unperformed), etc. In the system interface, a receiver (Expert User) points to this message to activate and open the server's Web page as shown in Figure 4, which will show this corresponding attributes of "Distribution Feed Bay", as follows:

**Bay No:** 12;  
**Substation Diagram File:** ChangWang.doc;  
**Subject:** Distribution Feed Bay;  
**Key Words of problem:** Please Check!  
**Shown Box:** Diagram on Distribution Feed Bay.

Then, the corresponding department of the system provider or expert users will provide an advice:

- If the problem is simple, the *expert user* may directly give an advice in *Advise ListBox*.
- If the problem is complicated, the *expert user* may directly hit the button "Get into the Application" to go into the shared software interface as in Figure 3, to modify the specified problem.

Finally, the *expert user* must hit the button "Inform User" to submit the solution to the client.

## 4 Conclusion

The collaborative management and training in the process of using the Smart Switching System is necessary for users, experts, and designers. In order to increase the efficiency and the quality of using the Smart Switching System and to reduce the costs of user company and system provider during the process of choosing a software application, a collaborative management and training model is proposed. A new collaborative environment based on the proposed model has been developed. By using the proposed model, a user company can evaluate the system and train its designers and operators. Normal users may visit the interactive Web page to submit training problems to the server created by the system provider. Expert users will then deal with the problems obtained from the Message Window. The proposed model paves

the way for the system provider and user companies to deal with the training of the Smart Switching System operators and provides a novel platform for further research in terms of collaborative management and training processes.

## Acknowledgement

The authors would like to thank the research group that took part in the study for their generous cooperation. The work presented in this paper was supported by Guangxi (China) Science Technology Subject under Grant No. 0330005-4C.

## References

1. Hollerer, T., Feiner, S., et al.: User interface management techniques for collaborative mobile augmented reality. *Computers & Graphics*, (25) (2001) 799-810
2. Peng, J. and Law, K.H.: A Prototype Software Framework for Internet-Enabled Collaborative Development of a Structural Analysis Program. *Engineering with Computers*, 18(1) (2002) 38-49
3. Tan, J.C, Liao, X.P., Wang, P.Z.: Substation Security Operating Design System Using Expert System Technology. 32nd Universities Power Engineering Conference, Manchester, UK, (1997) 381-384
4. Atick, J.: A future for ASPs? *Biometric Technology Today*, (8) (2001) 7-8
5. Egyedi, T. and Loeffen, A.: Editorial: XML diffusion: transfer and differentiation. *Computer Standards & Interfaces*, (24) (2002) 275-277
6. Liao, X., Liu, J., Zhang, X.: Graphic-Driven General Design Platform for Substation Operating System. *Journal of Computer-Aided Design & Computer Graphics*, 15(10) (2003) 1321-1328 (in Chinese).



# A Web-Based Fuzzy-AHP Method for VE Partner Selection and Evaluation

Jian Cao, Feng Ye, and Gengui Zhou

Institute of Information Intelligence and Decision-Making Optimization,  
Zhejiang University of Technology, Hangzhou, 310032, P.R. China  
jcao@iipc.zju.edu.cn

**Abstract.** In view of inconsistency of the judgment matrix obtained by pairwise comparison between increasing alternatives in virtual enterprise (VE) partner selection and evaluation problem by the Analytic Hierarchy Process (AHP), a new Web-based fuzzy-AHP method is proposed, in which the priority weights of decision criteria at every hierarchy are identified by the AHP and attribute-values of every alternative are determined by the fuzzy relation matrix, then they are converged to the solution. An prototype system based on this method has been developed, and a case study is used to demonstrate its practicability and effectiveness. The results indicate that the proposed method makes it easier for decision-makers to arrive at a consensus decision, obtain fair and reasonable conclusions and examine the strengths and weaknesses of alternatives in terms of each criterion. In addition, this method is suited to eliminating the drawbacks of existing traditional partner selection approaches or other AHP-based methods.

## 1 Introduction

With rapid development of information technology and economical globalization, the competition between companies has been changing from the quality and service of the product to the performance of the virtual enterprise (VE) in which the company is located. In the VE, the product is provided through the cooperation of all the partners from material supply to product marketing. Thus, even a small mistake taken place in one partner will slow down the response to the market and customers' demand. In order to improve or maintain the VE's competitive power, appropriate partners are very important.

The partner selection and evaluation is a very complex problem due to its inner multiple factors being interactive with each other. There has been rich literature on partner selection by the Analytic Hierarchy Process (AHP) or Fuzzy-AHP [1~7], but the approaches presented in [1~7] have a common problem [8]: when the number of alternatives is more than seven, the consistence ratio (CR) of corresponding judgment matrix is usually more than the rule-of-thumb value of CR equal to 0.1, hence evaluators' assessment bias will be obtained. However, if these alternatives are divided into several groups before evaluation (for instance, the number in each group is less than six), some good ones will be discarded in the beginning. Moreover, there were other problems such as negligence of considering group decision-making, personal judgment of

uncertain degree and complicated model structure in these approaches. Therefore, in order to improve the ability of solving today's VE partner selection problems, a new method combined the AHP with basic fuzzy theory is proposed and corresponding steps are described in this paper.

## 2 The VE Life Cycle

The VE is created to address a specific market opportunity quickly and concurrently, developing a common working environment to manage and use a collection of resources provided by the corresponding enterprises. The success of this particular mission depends on all enterprises cooperating as a synergetic unit during VE's whole life cycle. Generally, the life cycle of a new VE can be identified as five phases [9]: identification, formation, design, operation and dissolution. Following is the brief introduction of these phases:

- Identification phase: a core enterprise searches and recognizes the market opportunities, and then plans the formation of a new VE, estimating the costs and revenues of this potential new venture, the possible types of partnerships and B-B chains.
- Formation phase: the main task of the core enterprise during this phase is to find a number of potential organizations to perform tasks identified in the previous phase and to select the most suitable partners. Then, internal databases of all members are set up and coordinated through electronic communications.
- Design phase: the detailed procedures for carrying out the mission are specified during this phase, such as the design of new products and development of all material and information flows.
- Operation phase: the core enterprise schedules and synchronizes the partners' operational plans, monitors the progress of the mission and resolves the possible conflicts between the members of the VE.
- Dissolution phase: the main task during this phase is to archive the mission documentation, and to arrange after-sale services and customer support.

The key factor in forming the VE, emphasized by many researchers [7,10,11], is the selection of agile, competent and compatible partners. Partner selection is considered as a multiple criteria decision-making problem, and classical AHP method is widely used for solving such problems [12]. But with rapid development of Information Technologies, a new situation is created, in which the number of alternatives is increasing exponentially and it is becoming very difficult to filter candidates, thus many AHP-based methods mentioned in recent literature are not suited to solving partner selection and evaluation problem in the formation of the VE.

## 3 The Process of VE Partner Selection and Evaluation

In the past, partners were selected usually by the way of competitive bidding, which had many disadvantages such as small number of alternatives, regional restrictions, high operating costs, long evaluation and selection time, and inflexible processing way. Hence, the selected partners were usually not the best choices [13].

Rapidly developing Internet and increasingly perfecting E-Business environment provide a better way to select partners for the formation of VE. The core enterprise publishes its demands on its public Web server, and informs potential partners by way of email, telephone, mail, etc. Usually, these demands include the content of assessment, such as the product or service needed, date of delivery, the capability that potential partner should have and the deadline for response. After these demands are released through the Internet, the core enterprise begins to receive alternatives' responses. Web provides forms in which alternatives can fill corresponding information, which is automatically transformed and stored into the partner selection and evaluation database through CGI. Before the deadline, evaluators login the evaluation system and assess alternatives by prepared mathematical models and corresponding solution methods. When it reaches the deadline, the system will automatically select the best partner(s) and provide detailed evaluation report for reference of further decision-making.

## **4 A New Fuzzy-AHP Method for VE Partner Selection and Evaluation**

There are two main steps in the AHP method [14]: one is the determination of the normalized priority weights of criteria at every hierarchy, the other is the convergence of the normalized priority weights of criteria at the lowest hierarchy and attribute-values of each alternative to the evaluation result. If the number of criteria on which the same super-criterion is depended is more than seven, then divides them into several groups or reassembles them according to some common attributes. Because the priority weights of decision elements got by the AHP are generally reasonable and valid, it is a good way to identify those of decision criteria at every hierarchy by the AHP. But if the number of alternatives is more than seven, it is not suited for the AHP to determine each alternative's attribute-values, which in this paper mean alternative's subordinative degree to the bottom criteria of the decision hierarchy. Therefore, some basic fuzzy theory is combined with the AHP to determine the attribute-values of each alternative as well as the evaluation results of alternatives. The following subsections describe the four phases of this proposed fuzzy-AHP method.

### **4.1 Identifying the Decision Hierarchy and the Rating Set**

The decision problem is breaking down into a hierarchy of interrelated decision elements, as a tree containing the overall goal at the top with many levels of criteria and sub-criteria in between. One of the most important things is that the number of sub-criteria under the corresponding super-criterion should not be more than 6. Values in the rating set  $V$ ,  $V = \{v_1, v_2, \dots, v_m\}$ , are described by appropriate words, such as good, fair, poor, and so on. Each alternative will be assessed as one rating value in terms of every criterion. Generally, the number of rating values is odd, such as 3, 5, 7, which accords with quality requirement of fuzzy evaluation.

### 4.2 Estimating the Normalized Priority Weights of Decision Criteria

In the AHP, the solution  $\omega$ ,  $\omega=(\omega_1,\omega_2,\dots,\omega_n)^T$ , to Eq. (1) is called the priority weights vector of the logarithmic least square method (LLSM) [15]. After the normalization of  $\omega$ , the normalized priority weights vector of decision criteria  $w$ ,  $w=(w_1, w_2, \dots, w_n)^T$ , is obtained.

$$Z = \sum_{i=1}^n \sum_{j=1}^n [\ln a_{ij} - \ln(\frac{\omega_i}{\omega_j})]^2. \tag{1}$$

Taking different opinions of decision-makers into account, the pairwise comparison ratio  $a_{ij}$  in the judgment matrix  $A$  has more than one value. Therefore, by Eq. (1), we have

$$Z = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^b (\ln a_{ijk} - \ln \omega_i + \ln \omega_j)^2, \tag{2}$$

where  $b$  is a constant representing the number of decision-makers who take part in this group decision-making problem.

Take the partial derivative with respect to  $\omega_p$ ,  $p = 1, 2, \dots, n$ , for the function in Eq. (2), and let

$$\frac{\partial}{\partial \omega_p} \left[ \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^b (\ln a_{ijk} - \ln \omega_i + \ln \omega_j)^2 \right] = 0, \tag{3}$$

i.e.,

$$\sum_{\substack{k=1 \\ j \neq p}}^b \sum_{j=1}^n (\ln a_{pjk} - \ln \omega_p + \ln \omega_j) = \sum_{\substack{k=1 \\ i \neq p}}^b \sum_{i=1}^n (\ln a_{ipk} - \ln \omega_i + \ln \omega_p). \tag{4}$$

By Eq. (4), and noting that  $\ln a_{ipk} = \ln(1/a_{pjk})$ , it follows that

$$n \ln \omega_p - \sum_{j=1}^n \ln \omega_j = \frac{1}{b} \sum_{j=1}^n \sum_{k=1}^b \ln a_{pjk}, \tag{5}$$

in other words,

$$\omega_p = \left( \prod_{k=1}^b \prod_{j=1}^n \omega_j \right)^{\frac{1}{bn}} \cdot \left( \prod_{k=1}^b \prod_{j=1}^n a_{pjk} \right)^{\frac{1}{bn}}. \tag{6}$$

Finally, normalizing  $\omega$  to  $w$ , we obtain

$$w_p = \left( \prod_{k=1}^b \prod_{j=1}^n a_{pjk} \right)^{\frac{1}{bn}} / \sum_{i=1}^n \left( \prod_{k=1}^b \prod_{j=1}^n a_{ijk} \right)^{\frac{1}{bn}}. \tag{7}$$

### 4.3 Assessing Alternatives and Identifying Corresponding Fuzzy Relation Matrix

Let  $R$  be the fuzzy relation matrix [16] of the alternative. Firstly, each alternative is quantified in terms of every bottom criterion  $u_i, i=1, 2, \dots, n$ . Then subordinative degree ( $R|u_i$ ) to each alternative is determined. Therefore, we get the following equation

$$R = \begin{bmatrix} R|u_1 \\ R|u_2 \\ \dots \\ R|u_n \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1m} \\ r_{21} & r_{22} & \dots & r_{2m} \\ \dots & \dots & \dots & \dots \\ r_{n1} & r_{n2} & \dots & r_{nm} \end{bmatrix}_{n \times m}, \tag{8}$$

where  $r_{ij}$  is alternative’s subordinative degree to the bottom criterion  $u_i$  in terms of the rating value  $v_j, j=1, 2, \dots, m$ . Taking personnel promotion as an example, ten decision-makers evaluate three potential manager candidates (namely A, B and C). Three decision criteria,  $u_1 =$  ability,  $u_2 =$  morality and  $u_3 =$  health, are proposed. And there are three rating values,  $v_1 =$  good,  $v_2 =$  average and  $v_3 =$  poor in the rating set. To evaluate A’s ability, if four decision-makers consider “good”, five “average” and one “poor”, we will get

$$R_A | u_1 = (0.4, 0.5, 0.1). \tag{9}$$

### 4.4 Calculating the Evaluation Result Vector of Each Alternative and Synthesizing the Solution

Let  $S$  be the evaluation result vector [16], which represents subordinative degree of certain alternative in terms of the whole rating set,

$$S = w \circ R = (w_1, w_2, \dots, w_n) \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1m} \\ r_{21} & r_{22} & \dots & r_{2m} \\ \dots & \dots & \dots & \dots \\ r_{n1} & r_{n2} & \dots & r_{nm} \end{bmatrix} = (s_1, s_2, \dots, s_m), \tag{10}$$

where  $\circ$  represents fuzzy composite operator  $M(\cdot, \oplus)$ , for  $j=1, 2, \dots, m$ ,

$$s_j = (w_1 \cdot r_{1j}) \oplus (w_2 \cdot r_{2j}) \oplus \dots \oplus (w_n \cdot r_{nj}) = \min(1, \sum_{i=1}^n w_i r_{ij}) = \min(1, \sum_{i=1}^n w_i r_{ij}) \tag{11}$$

Finally, after processing the evaluation result vector of each alternative by weighted mean method, the resulting value  $T$ , which represents alternative’s relative position in the rating set, is obtained,

$$T = \left( \sum_{j=1}^m s_j^k \cdot j \right) / \left( \sum_{j=1}^m s_j^k \right), \tag{12}$$

where  $k$  is an undetermined coefficient, generally  $k=2$ . Obviously, the less  $T$ , the better the alternative.

### 5 An Example

According to the above-mentioned programming method, a Web-based VE partner selection and evaluation system is developed. The main menu of the system consists of alternatives, evaluators, criteria, evaluators' opinions and evaluation results.

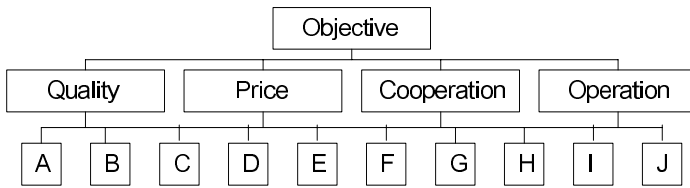


Fig. 1. A decision hierarchy of supplier selection

Combined with a group decision-making for the selection of lumber suppliers in a large-scale paper mill, the application of the system is demonstrated in this section. As shown in Fig.1, three-hierarchy structure is formulated in this example. Four main criteria including quality of lumber, price of lumber, cooperation and operation of supplier have been chosen during the identification phase, and ten suppliers, namely from A to J, have been identified as potential partners. The goal here is to select two partners, satisfying all criteria in the best way. Main steps for evaluating and selecting VE partners are as follows:

Step 1. Determining evaluators who qualified for the decision-making of this problem.

Three teams, one including three operating managers in the supplying department, one including four senior managers in the paper mill and the other including three professional members outside the mill, totally ten evaluators take part in this partner selection problem.

Step 2. Determining the normalized priority weights of the lowest criteria.

By assigning nine-point scale pairwise comparisons to four criteria, three teams give the judgment matrix (13), from which, combined with Eq. (7), the normalized priority weights of these criteria are obtained in Table 1.

$$\begin{matrix}
 & \begin{matrix} \text{Quality} & \text{Price} & \text{Cooperation} & \text{Operation} \end{matrix} \\
 \begin{matrix} \text{Quality} \\ \text{Price} \\ \text{Cooperation} \\ \text{Operation} \end{matrix} & \begin{bmatrix} (1,1,1) & (1,1,3/2) & (2,3/2,2) & (3/2,1,2) \\ (1,1,2/3) & (1,1,1) & (3/2,3/2,1) & (3/2,3/2,1) \\ (1/2,2/3,2/1) & (2/3,2/3,1) & (1,1,1) & (3/2,1,4/3) \\ (2/3,1,1/2) & (2/3,2/3,1) & (2/3,1,3/4) & (1,1,1) \end{bmatrix}
 \end{matrix} \tag{13}$$

**Table 1.** The normalized priority weights of four criteria

	Criterion			
	Quality	Price	Cooperation	Operation
Normalized priority weight	0.3225	0.2712	0.2090	0.1973

Step 3. Assessing each alternative in terms of every lowest criterion and identifying corresponding fuzzy relative matrix.

**Table 2.** Values in the rating set

	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$
Corresponding value	Outstanding	Good	Fair	Poor	Unacceptable

Firstly, we determine the rating set  $V$  of this problem,  $V=\{v_1, v_2, \dots, v_5\}$ , and values in the set are shown as Table 2. Then, after synthesizing ten evaluators' opinions on each alternative in terms of every criterion, fuzzy relative matrix of each alternative is determined. For example, fuzzy relative matrices of supplies A, B, C and D are as follows, respectively:

$$R_A = \begin{bmatrix} 0.3 & 0.4 & 0.3 & 0 & 0 \\ 0 & 0.2 & 0.5 & 0.3 & 0 \\ 0.2 & 0.7 & 0.1 & 0 & 0 \\ 0.1 & 0.3 & 0.4 & 0.1 & 0.1 \end{bmatrix}. \tag{14}$$

$$R_B = \begin{bmatrix} 0.2 & 0.2 & 0.5 & 0.1 & 0 \\ 0.3 & 0.5 & 0.2 & 0 & 0 \\ 0 & 0.1 & 0.7 & 0.2 & 0 \\ 0 & 0.2 & 0.5 & 0.3 & 0 \end{bmatrix} \tag{15}$$

$$R_C = \begin{bmatrix} 0 & 0.1 & 0.6 & 0.3 & 0 \\ 0.3 & 0.5 & 0.2 & 0 & 0 \\ 0.4 & 0.5 & 0.1 & 0 & 0 \\ 0 & 0.2 & 0.5 & 0.3 & 0 \end{bmatrix} \tag{16}$$

$$R_D = \begin{bmatrix} 0.4 & 0.5 & 0.1 & 0 & 0 \\ 0 & 0.2 & 0.3 & 0.4 & 0.1 \\ 0.1 & 0.5 & 0.3 & 0.1 & 0 \\ 0.3 & 0.6 & 0.1 & 0 & 0 \end{bmatrix} \tag{17}$$

Step 4. Calculating the evaluation result vector of each alternative and accessing priorities.

By Eq. (10), we get each alternative’s evaluation result vector  $S_X$  ( $X$  represents alternative), which is shown in Table 3. We can obtain some useful information from  $S_X$ , for example, in terms of rating value  $v_1$  (i.e., “outstanding”), alternative I is the best and F the worst. By Eq.(12), the resulting value  $T_X$  is also obtained, thus the ranking of each alternative is determined, as shown in Table 3, from which we get the best two suppliers: I and G.

**Table 3.** The evaluation result vector  $S_X$  and the resulting value  $T_X$  ( $X$  represents alternative)

Alternatives	The evaluation result vector, $S_X$	The resulting value, $T_X$	Ranking
A	(0.1583, 0.3887, 0.3322, 0.1011, 0.0197)	2.3598	5
B	(0.1459, 0.2605, 0.4604, 0.1332, 0)	2.7093	9
C	(0.1650, 0.3118, 0.3673, 0.1559, 0)	2.5510	7
D	(0.2091, 0.4384, 0.1960, 0.1294, 0.0271)	2.0975	3
E	(0.1701, 0.3628, 0.3350, 0.1123, 0.0197)	2.3838	6
F	(0.0542, 0.3225, 0.4040, 0.1840, 0.0323)	2.7528	10
G	(0.2888, 0.5000, 0.1914, 0.0197, 0)	1.8758	2
H	(0.1040, 0.3446, 0.4504, 0.1011, 0)	2.6201	8
I	(0.2904, 0.6084, 0.1012, 0, 0)	1.8406	1
J	(0.1112, 0.5572, 0.2848, 0.0469, 0)	2.1801	4

This simplified example is chosen only for a better understanding of the main principles of the proposed method. In real situations the number of criteria and alternatives could be greater, and the decision hierarchy might include intermediate levels of sub-criteria. However, this method can be easily applied for prioritization problems of greater dimensions, since it requires solving some simple linear programs. From a computational point of view this is not a problem, since the standard Simplex method can be easily used to solve problems with hundreds and thousands of variables and constraints [17].

## 6 Conclusions

Combined qualitative analysis with quantitative analysis efficiently, the AHP was once a powerful method to the solution of traditional partner selection and evaluation problem. But with rapidly developing Internet and increasingly perfecting E-Business environments, more and more potential partners are short-listed in the formation of VE nowadays. Due to inconsistency of the judgment matrix obtained by pairwise comparison between alternatives, the AHP is not well suited to obtaining the attribute-values of alternative now. Therefore, Combined with other basic fuzzy theory, a new fuzzy-AHP method for VE partner selection and evaluation is proposed. It is manifested that this method is not only suited to eliminating drawbacks including limited number of alternatives, personal judgment of uncertain degree and unreasonable



mathematical model structure, which are caused by other AHP-based methods, but also suited to solving problems such as regional restrictions, long treating time and inflexible processing way, which are produced by some traditional partner selection approaches. By using this proposed method, impartial and reasonable results are easier to be obtained, and other helpful information is also easier to be gotten from the evaluation result vector of every alternative to facilitate decision-making.

## Acknowledgement

This research work was partially supported by Research Planning Fund of Zhejiang Provincial Education Department (No.20040580) and Zhejiang Provincial Nature Science Foundation (No.Y104171).

## References

1. Maggie, C.Y., Rao, V.M.: An Application of the AHP in Vendor Selection of a Telecommunications System. *Omega*. 29 (2001) 171-182
2. Wang, J.S., Wang, T.M., Hu, Y.G.: Study of a Supplier Evaluation Model with Fuzzy Analytic Hierarchy Process. *Microelectronics and Computer*. 18 (2001) 59-64
3. Yu, C.S.: A GP-AHP Method for Solving Group Decision-Making Fuzzy AHP Problems. *Computer and Operation Research*. 29 (2002) 1969-2001
4. Wang, R.X., Ruo, Q., Zhu, M.Q.: Study on Partners Selection Modeling in Virtual Enterprise. *Chinese Journal of Mechanical Engineering*. 38 (2002) 28-30
5. Ossadnik, W.: AHP-based Synergy Allocation to Partners in a Merger. *European Journal of Operational Research*. 88 (1996) 42-49
6. Nydick, R., Hill R.: Using the AHP to Structure the Supplier Selection Procedure. *Journal of Purchasing and Materials Management*. 25 (1992) 31-36
7. Talluri, S., Baker, R., Sarkis, J.: A Framework for Designing Efficient Value Chain Networks. *International Journal on Production Economics*. 62 (1999) 133-144
8. Liu, X.X.: *Selection and Judgment*. Shanghai Popularization Science Press, Shanghai (1990)
9. Kanet, J.: Application of IT to a Virtual Enterprise Broker. *International Journal on Production Economics*. 62 (1999) 23-32
10. Jagdev, H., Browne, J.: The Extended Enterprise -- a Context for Manufacturing. *Production Planning and Control*. 9 (1998) 216-229
11. Papazoglou, M., Ribbers, P.: Integrated Value Chains and Their Applications Form a Business and Technology Standpoint. *Decision Support Systems*. 29 (2000) 323-342
12. Meada, L., Liles, D., Sarkis, J.: Justifying Strategic Alliances and Partnering: a Prerequisite for Virtual Enterprise. *Omega*. 25 (1997) 29-42
13. Chen, J.H., Wang, Y.L., Shun, L.Y.: Study on the Processes and Methods of Partner Selection in Virtual Organization. *Systems Engineering --Theory and Practice*. 27 (2001) 48-53
14. Saaty, T.L.: *The Analytic Hierarchy Process*. McGraw-Hill, New York (1980)
15. Grawford, G., Williams, C.A.: A Note on the Analysis of Subjective Judgment Matrices. *Journal of Mathematical Psychology*. 29 (1985) 387-405
16. Hu, Y.H., He, S.H.: *Synthetic Evaluation Methods*. Science Press, Beijing (2000)
17. Nebojsa, V.: Two Direct Methods in Linear Programming. *European Journal of Operational Research*. 131 (2001) 417-439

# A Method of Network Simplification in a 4PL System

He Zhang<sup>1</sup>, Xiu Li<sup>1</sup>, and Wenhua Liu<sup>2</sup>

<sup>1</sup> National Engineering Research Center for CIMS, Dep. of Automation,  
Tsinghua University, 100084 Beijing, China  
zhanghe98@mails.tsinghua.edu.cn,  
lixiu@cims.tsinghua.edu.cn

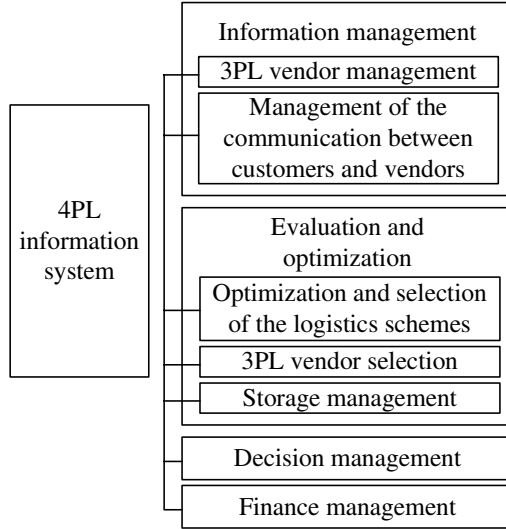
<sup>2</sup> Graduate School at Shenzhen, Tsinghua University,  
518055 Shenzhen, China  
liuwh@sz.tsinghua.edu.cn

**Abstract.** In fourth party logistics (4PL), network optimization needs to synthesize the entire possible 3PL vendors to provide integrated logistics schemes, which can meet all the requirements of customers. Since there are knapsack constraints and un-knapsack constraints, the computational burden is considered to be a problem and a modest computation is required. In this paper, a 4PL optimization network-model is established and a two-process method is suggested for simplifying the corresponding 4PL network. An example is provided and some analysis is given after the calculation.

## 1 Introduction

In today's industrial scenario, suppliers and big corporations need to meet increased levels of services due to e-procurement, complete supply visibility, virtual inventory management and requisite integrating technology. Corporations are outsourcing their entire set of supply chain process from a single organization which will assess, design, make and run integrated comprehensive supply chain solutions [1,2]. The evolution in supply chain outsourcing is called fourth party logistics (4PL)— the aim being to provide maximum overall benefit.

A 4PL decision support model is shown in Figure 1 [3-5]. As shown in this figure, network optimization is an important problem in 4PL. A 4PL is based on third party logistics (3PL) vendors. A 4PL vendor should integrate the information of all the 3PL vendors, establish a directed network and search a comprehensive solution. An optimized path should be selected in the network according to the customers' requirements. Thus network optimization is an important problem. In a directed network, each arc has several attributes such as cost, time and length. In 4PL, the problem is to seek the path between two certain nodes that minimize one attribute and to satisfy constraints on other attributes. In the problem, there are two kinds of constraints, un-knapsack and knapsack. Un-knapsack constraints, such as logistics capacity, require each arc to meet the requirement. However, in one path, the sum of all the relevant attributes should meet the requirement of the knapsack constraints, such as time and cost.



**Fig. 1.** A framework of a 4PL decision support system. The part of evaluation and optimization is the key problem in the 4PL system.

The computational burden of the problem is a major obstacle that can prohibit the application of the relevant algorithms. It is possible that a node or an arc would not in any feasible path. So if these unfeasible nodes and arcs can be removed from the directed network, the computational burden can be reduced for a better efficiency.

Knapsack constraints are the main factors to be considered to reduce the computational burden [6], because un-knapsack constraints can be met by reduce a specific arc, whose relevant attributes cannot meet the requirements. In order to alleviate the computational burden, a modification has been suggested by Skiscikm and Golden [7]. Hassan [8] presented a procedure to reduce the computational burden. Some methods were proposed to use Lagrange factors [6,9] to reduce this burden.

In a 4PL system, customers often have many requirements [10,11] and they also need many logistics schemes for further selections [12]. Because of the burden caused by these corresponding operating conditions, it is important for the solution to obtain a modest computation.

In this paper, a 4PL optimization network-model is established and a method is suggested for reducing the acyclic networks with constraints. The method can be applied to such a network before applying one optimization algorithm.

## 2 Modeling the Optimization Problem

With the relevant information of 3PL vendors, a network optimization model with multi-dimensional weigh is established.

Cities (start city, destination city, transferring cities) are corresponding to nodes in the network and are denoted as  $v_{label}$ , *label* can be start, destination, or the index of the transferring cities. If there is a 3PL provider supplying logistics from city  $i$  to city

$j$ , a relevant arc is drawn from the corresponding node  $i$  to node  $j$  in the network. The attribute of the 3PL vendor is defined as the multi-dimension weight bound to the relevant arc denoted as  $a_{ij(k)}$ . The multi-dimensional weight is composed of four parts, which are logistics capacities information, logistics time information, logistics cost information, and 3PL evaluation information. The weight is a vector defined as  $(C, T, B, E)$ .

$C$  is the information about cost and also is a vector:

$$(price\ 1, unit\ 1; \dots; price\ n, unit\ n)$$

$T$  is the information vector about logistics time.

$B$  contains the necessary information about the 3PL provider’s logistics ability.

$E$  is the evaluation information about 3PL vendor (logistics throughput capacity, logistics technology, vendor’s reputation, etc)

Obviously, constraints on  $C$  and  $T$  are knapsack ones. While those on  $B$  and  $E$  are un-knapsack factors. In this paper, the attributes  $C$ ,  $T$ ,  $B$  and  $E$  are all one-dimension variables.

According to the above conventions, the model of the directed network optimization in 4PL is defined in the following form:

$$\left\{ \begin{array}{l} V = \{v_1, v_2, \dots, v_i, \dots, v_N\} \\ A = \{a_{12(1)}, a_{12(2)}, \dots, a_{ij(k)}, \dots\} \\ P = \{p_{st} = (a_{si_2(j_1)}, \dots, a_{i_{m-1}t(j_{m-2})}) \mid a_{si_2(j_1)}, \dots, a_{i_{m-1}t(j_{m-2})} \in A\} \\ F(p_{st}^*) = \min F(p_{st}) \\ G_l(p_{st}^*) > 0, (l = 1, 2, \dots, h) \end{array} \right.$$

$V$  is the node set and each node represents a specific city.  $A$  is the arc set and each arc represents a specific path, which connects two nodes— two specific cities.  $a_{ij(k)}$  represents the  $k$ th arc, one 3PL provider, which links  $v_i$  and  $v_j$ .  $P$  is the feasible path set and each  $p_{st}$  presents a specific feasible path from the source node  $s$  to sink node  $t$ .  $F$  is the objective function set, which represents the optimizing objects, such as the least cost, the shortest time and etc.  $G$  is the bound function set, which represents the restraint factors, such as the proper cost, limited time and etc.

The constraints are supposed as:

$$C \leq C_{\max}, T \leq T_{\max}, B \geq B_{\min}, E \geq E_{\min}$$

$C_{\max}$ ,  $T_{\max}$ ,  $B_{\min}$  and  $E_{\min}$  are bound constraints according to the customers’ requirements.

### 3 Network Simplification

#### 3.1 Definitions and Concepts

The algorithm is to identify the arcs and nodes that cannot be on any feasible path for the given bound constants of  $C_{\max}$ ,  $T_{\max}$ ,  $B_{\min}$  and  $E_{\min}$ . It is accomplished by traveling the network twice using arc attributes  $(C, T, B, E)$ . The arcs and nodes that cannot be on any feasible path are referred to as useless arcs (UAs) and useless nodes (UNs). In these two processes, cost and time labels are established as in shortest path algorithms of the label setting type.

#### 3.2 Forward Traveling

In this step, the network is traversed forward starting at node  $s$ . Several label-vectors are established for node  $i$ , each representing the relevant attributes from the source node  $s$  to node  $i$  via a particular path. Let  $N_i$  be the set of nodes having arcs entering node  $i$ .  $M_i$  is the set of the arcs entering node  $i$ . There are  $|M_i|$  label-vectors for each arc from node  $i$  to the node in  $N_i$ . Each is found as:

$$W'_{ki(j)} = [u_{ki(j)}, w_{ki(j)}, x_{ki(j)}, y_{ki(j)}]^T, v_k \in N_i, a_{ki(j)} \in M_i, i = 2, \dots, N$$

$$u_{ki(j)} = u_k + c_{ki(j)}, w_{ki(j)} = w_k + t_{ki(j)}, x_{ki(j)} = b_{ki(j)}, y_{ki(j)} = e_{ki(j)}$$

$(c_{ki(j)}, t_{ki(j)}, b_{ki(j)}, e_{ki(j)})$  is the relevant attribute-vector of arc  $a_{ki(j)}$ .  $(u_k, w_k)$

is the permanent label-vector of node  $v_k$ ,  $v_k \in N_i$ .

Obviously, if  $v_0 = s$ ,  $(u_0, w_0) = 0$ . The label-vectors of node  $i$  are used to calculate  $W_{ik(j)}$  and  $W_{ik(j)}$  is defined as following:

$$F_{ki(j)} = C_{\max} - u_{ki(j)}, G_{ki(j)} = T_{\max} - w_{ki(j)}$$

$$W_{ik(j)} = [F_{ki(j)}, G_{ki(j)}, x_{ki(j)}, y_{ki(j)}]^T, v_k \in N_i, a_{ki(j)} \in M_i$$

**Proposition 1.** If  $x_{ki(j)} < B_{\min}$  or  $y_{ki(j)} < E_{\min}$ , arc  $a_{ki(j)}$  is a UA.

**Proof.** This proposition is obviously right for the two constraints are un-knapsack.

Let:  $M_i = \{a_{ki(j)} \mid x_{ki(j)} \geq B_{\min}, y_{ki(j)} \geq E_{\min}, a_{ki(j)} \in M_i\}$

The permanent label-vector for node  $i$  is calculated as following:

$$D_i = [u_i, w_i]^T, u_i = \min_{a_{ki(j)} \in M_i} \{u_{ki(j)}\}, w_i = \min_{a_{ki(j)} \in M_i} \{w_{ki(j)}\}$$

The upper bounds on the remaining cost and time at node  $i$  are established from the values of  $F_{ki(j)}$  and  $G_{ki(j)}$

$$F_i = \max_{a_{ki(j)} \in M_i} \{F_{ki(j)}\} = C_{\max} - u_i, G_i = \max_{a_{ki(j)} \in M_i} \{G_{ki(j)}\} = T_{\max} - w_i$$

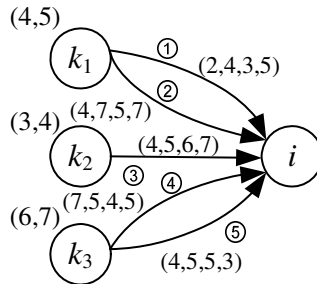
**Proposition 2.** If  $F_k > 0$  and  $G_k > 0$ , when  $F_{ki(j)} \leq 0$  or  $G_{ki(j)} \leq 0$ , arc  $a_{ki(j)}$  is a UA.

**Proof.** If  $F_k > 0$  and  $G_k > 0$ , then there must be a feasible path from source node  $s$  to node  $k$ . Further,  $F_{ki(j)} \leq 0$  is equivalent to  $F_k - c_{ki(j)} < 0$ . Thus, the cost required to reach node  $i$  from node  $k$  is greater than the largest cost available at node  $k$  and arc  $a_{ki(j)}$  is a UA.  $G_{ki(j)} \leq 0$  is in like manner.

**Corollary 2.1.** If  $F_i \leq 0$  or  $G_i \leq 0$ ,  $v_i \neq t$ , node  $i$  is UN.

**Proof.**  $F_i = \max_{a_{ki(j)} \in M_i} \{F_{ki(j)}\}$ . If  $F_i \leq 0$ , then all  $F_{ki(j)} \leq 0$ , and all arcs  $(k, i)$ , are UAs according to Proposition 1. Consequently, all the arcs leaving node  $i$  cannot be used. Thus, node  $i$  is a UN.

An illustration of the above situation is given in Figure 2.



**Fig. 2.** Identifying UAs & UNs in forward traveling. In this figure, Labels of nodes are  $(F_m, G_m)$ , labels of arcs are  $(c_{ki(j)}, t_{ki(j)}, b_{ki(j)}, e_{ki(j)})$ . Arc ① and arc ⑤ are two UAs according Proposition 1. Arc ②, arc ③ and ④ are three UAs according to Proposition 2. Node  $i$  is a UN according to Corollary 2.1.

It should also be pointed out that if at the end of the forward traveling,  $F_t < 0$  or  $G_t < 0$ , then the problem is infeasible.

### 3.3 Backward Traveling

In the second step of the procedure, labels are established for the nodes while traversing the network backward. That is, node  $i$  will be labeled from node  $j$  as if  $a_{ij(k)}$  were  $a_{ji(k)}$ . Let  $J_i$  be the set of nodes at which the arcs leaving node  $i$  end and  $H_i$

be the relevant set of arcs. Thus there are  $|H_i|$  labels for node  $i$  from the nodes in set  $J_i$ , each is calculated as:

$$L_{ji(k)} = \left[ u'_{ji(k)}, w'_{ji(k)} \right]^T, v_j \in J_i, a_{ij(k)} \in H_i, i = 2, \dots, N$$

$$u'_{ji(k)} = u'_j + c_{ij(k)}, w'_{ji(k)} = w'_j + t_{ij(k)},$$

$(u'_j, w'_j)$  is the permanent label-vector of node  $v_j$ .  $u'_i$  and  $w'_i$  are determined as:

$$u'_i = \min_{a_{ij(k)} \in H_i} \{u'_{ji(k)}\}, w'_i = \min_{a_{ij(k)} \in H_i} \{w'_{ji(k)}\}$$

Obviously,  $u'_i = 0, w'_i = 0$

$u'_i$  represents the smallest cost that can be used to traverse the network from node  $i$  to the sink node  $t$ .  $w'_i$  represents the smallest time that can be used to traverse the network from node  $i$  to the sink node  $t$ . The labels of node  $i$ , established in the backward traveling,  $u'_i$  and  $w'_i$  are used in conjunction with  $F_i$  and  $G_i$  to identify whether one arc entering or leaving node  $i$  is a UA and whether node  $i$  itself is a UN according to the following proposition.

**Proposition 3.** If  $u'_{ji(k)} > F_i$  or  $w'_{ji(k)} > G_i, a_{ij(k)} \in H_i$ , then arc  $a_{ij(k)}$  is a UA.

**Proof.** Since  $F_i$  is the largest cost remaining from node  $i$  to sink node  $t$ . So if  $u'_{ji(k)} > F_i$ , there is no valid path, which traverse node  $i$  and arc  $a_{ij(k)}$  at the same time. Consequently, arc  $a_{ij(k)}$  is a UA.  $w'_{ji(k)} > G_i$  is in like manner.

**Corollary 3.1.** If  $u'_i > F_i$  or  $w'_i > G_i$ , then node  $i$  is a UN since none of the arcs leaving it, and consequently those entering it, can be used.

**Corollary 3.2.** If  $u'_i > F_{ki(j)}$  or  $w'_i > G_{ki(j)}$ , arc  $a_{ki(j)}$  is a UA.

In the backward traveling, the labels do not include similar items like the labels— $x_{ki(j)}, y_{ki(j)}$  in the forward traveling. This is because all the UAs, which do not meet the requirement of the attributes  $B$  and  $E$  have been deleted in the forward traveling according to Proposition 1.

**Proposition 4.** For node  $i$  and node  $j$ , if  $F_i > 0, G_i > 0, F_j > 0, G_j > 0$ , and  $u_j = u_i + t_{ij(k)}$ , then node  $i$  cannot be a UN in the backward traveling.

**Proof.** Since node  $j$  is not a UN, then  $u'_j \leq F_j$  and  $w'_j \leq G_j$ . If  $u'_j \leq F_j$ , then  $u'_i \leq F_j$ . So

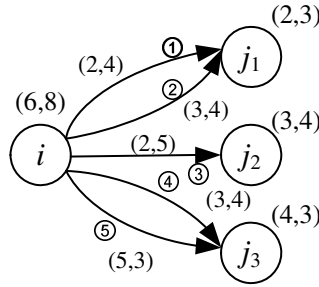
$$u'_j + t_{ij(k)} \leq F_j + t_{ij(k)}, \text{ then } u'_{ji(k)} \leq F_j + t_{ij(k)}.$$

$$\text{So } u'_{ji(k)} \leq C_{\max} - u_j + t_{ij(k)} = C_{\max} - u_i = F_i$$

Thus  $u'_i \leq u'_{ji(k)} \leq F_i$ , then node  $i$  cannot be a UN in the backward traveling.

According to Proposition 4, we can find that once a node is used to label another node permanently in the forward pass, its labels,  $u_i$  and  $w_i$  cannot be changed as removing an arc or a node in the backward traveling.

The illustration of the above situations is given in Figure 3.



**Fig. 3.** Identifying UAs & UNs in backward traveling. In this figure, Labels of node  $i$  are  $(F_i, G_i)$ , labels of arcs are  $(c_{ij(k)}, t_{ij(k)})$ . All the arcs are UAs according Proposition 3. Node  $i$  is a UN according to Corollary 3.1.

### 3.4 Remarks

In such a simplified problem, one attribute,  $F_i$  or  $G_i$ , often needs to be optimized. So there are often only three constraints such as,  $T \leq T_{\max}$ ,  $B \geq B_{\min}$ ,  $E \geq E_{\min}$  and one optimized objective, such as  $\min C_{path}$ . Thus when applying the simplification method, there are only one knapsack constraint.  $T \leq T_{\max}$  or  $C \leq C_{\max}$ .

It should be pointed out that in the forward traveling, the labels,  $F_i$  or  $G_i$  may be calculated from different arcs, even different nodes. Thus it is possible that a certain node  $k_1$ , which ends node  $i$ , is used to calculate the permanent label  $F_i$ , while it is a UN according to Proposition 2. So when calculating the permanent labels of node  $i$ , all the UAs according to Proposition 1 and Proposition 2.

In this simplification method, the network should be traversed first in the forward traveling rather than in the backward traveling. The upper bounds established in the forward traveling are to ensure the availability of all the four constraints at each node after reaching it. In the backward traveling, the upper bounds established in the for-



ward traveling are tested against the lower bound in order to ensure that it is sufficient to traverse a path from this node to the end node.

### 4 Steps of the Method

As there is usually only one knapsack constraint in the problem. So when introducing the steps of the method,  $T \leq T_{\max}$ ,  $B \geq B_{\min}$  and  $E \geq E_{\min}$  is supposed to be the objective constraints. The steps of the suggested method may be summarized as following:

Step 1. Traverse the network forward using the attributes  $(T, B, E)$  and the relevant constraints. This procedure is shown in Figure 4 (a).

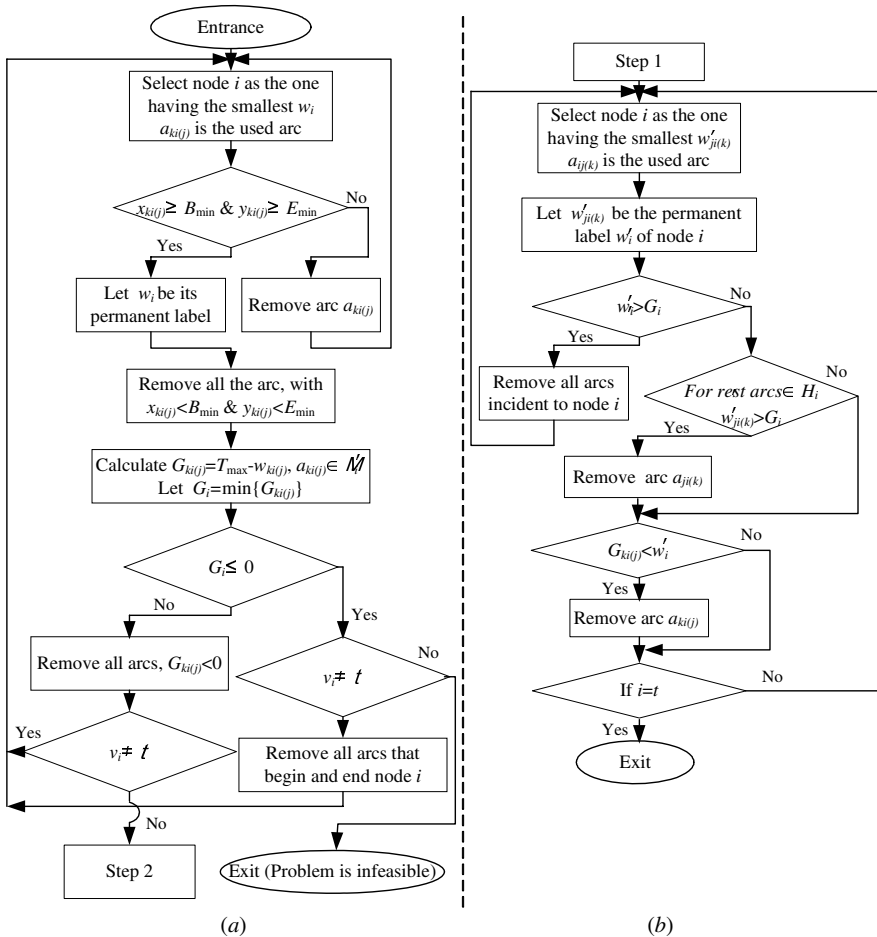
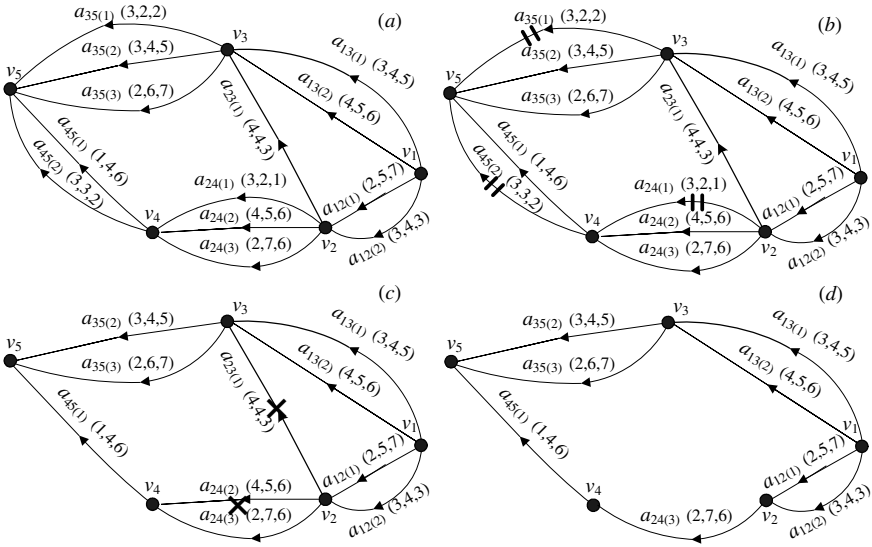


Fig. 4. Procedure of the method. (a) procedure of the forward traveling, (b) procedure of the backward traveling.

Step 2. Traverse the network backward using attribute  $T$ . This procedure is shown in Figure 4 (b).

### 5 Computation Example

A example is solved for  $T_{\max} = 6$ ,  $B_{\min} = 3$  and  $E \geq 2$ , and is illustrated in Figure 5.



**Fig. 5.** Procedure of simplifying a specific problem. The attributes on each arc are  $(T, B, E)$ . (a) the original optimization problem; (b) the simplified problem after the forward traveling; (c) the simplified problem after the backward traveling; (d) the final simplified problem.

In the forward traveling, arcs  $a_{24(1)}$  and  $a_{35(1)}$  are UAs according to Proposition 1, because arc  $a_{24(1)}$  cannot meet the constraint  $E \geq 2$  and arc  $a_{35(1)}$  cannot meet the constraint  $B_{\min} = 3$ . Arc  $a_{45(2)}$  is a UA according to Proposition 2, because the max time provided by node  $v_4$  is 2. In the backward traveling, arc  $a_{23(1)}$  is a UA according to Proposition 3. Arc  $a_{24(2)}$  is a UA according to Corollary 3.2. Thus it can be pointed out that a whole simplification procedure should include both a forward traveling process and a backward traveling process. The forward traveling process should be applied before the backward traveling.

## 6 Conclusion

In this paper, a two-process method is suggested for 4PL network simplification. The model of the network is established and the problem is described as an optimization with four constraints, including knapsack and un-knapsack ones. The computations required for the method are modest and involve establishing arc and node labels, computing upper bounds and identifying UAs and UNs. The most important merit of the method is that it can remove most nodes and arcs that cannot appear in any feasible path.

It should be pointed out that when there are two knapsack constraints, the network should be traveled for four times, two forward travels and two backward travels because of the difference of the two labels.

## Acknowledgement

This work presented in this paper is supported by the National Science Foundation of China (Grant No. 70202008).

## References

1. Shen, S.: The Analysis Report of the Supply and Demand Status of Chinese Logistics Market. *Logistics Management*, 2 (2000) 3–14
2. Xu, J.: New Pattern of Supply Chain Management—the Fourth Distribution Channel. *Policy-making Reference*, 15(1) (2002) 11–15
3. Chen, J., Liu, W., Zhang, A.: The Exploring of the Fourth Party Logistics and Decision Support. *Science of Science and Management of S&T*, 23(9) (2002) 72–74
4. Zhang, H., Li, X., Liu, W., Li, B., Zhang, Z.: An Application of the AHP in 3PL Vendor Selection of a 4PL System. *Proceedings of the IEEE International Conference on Systems, Man & Cybernetics*, Hague, (2004) 1255–1260
5. Zhang, H., Liu, W., Li, X.: Appraisalment of Transporters in Fourth Party Logistics. *Industrial Engineering Journal*, 7(3) (2004) 36–39
6. Handler, G.Y., Zang, I.: A Dual Algorithm for the Constrained Shortest Path Problem. *Networks*, 10(4) (1980) 293–310
7. Skiscim, C.C., Golden, B.L.: Solving K-Shortest and Constrained Shortest Path Problems Efficiently. *Annals of Operations Research*, 20 (1996) 249–282
8. Hassen, M.D.: Network Reduction for the Acyclic Constrained Shortest Path Problem, *European Journal of Operational Research*, 63(1) (1992) 124–132
9. Aneja, Y.P., Nair, K.P.K.: The Constrained Shortest Path Problem. *Naval Research Logistics Quarterly*, 25 (1978) 549–555
10. Chen, J., Wang, S., Li, X., Liu, W.: Directed Graph Optimization Model and Its Solving Method Based on Genetic Algorithm in Fourth Party Logistics. *Proceedings of the IEEE International Conference on Systems, Man & Cybernetics*, Washington, D.C., (2003) 1961–1966
11. Lau, H.C., Goh, Y.G.: An intelligent brokering system to support multi-agent web-based 4th party logistics. *Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence*, Washington, D.C., (2002) 154–161
12. Chen, J., Liu, W., Li, X.: The Directed Graph Model with Multi-Dimensions in the Fourth Party Logistics and its Algorithm. *Industrial Engineering and Management*, 8(3) (2003) 45–49

# Using Augmented Reality Technology to Support the Automobile Development

Jürgen Fründ, Jürgen Gausemeier, Carsten Matyszcok, and Rafael Radkowski

Heinz Nixdorf Institute, University of Paderborn,  
Fürstenallee 11, 33106 Paderborn, Germany  
{juergen.fruend, juergen.gausemeier, carsten.matyszcok,  
rafael.radkowski}@hni.upb.de

**Abstract.** Today a high-quality automotive design is one essential factor for the success of a new automobile. The increasing safety, comfort and communication functionalities require coherent visibility and design and handling concepts to avoid irritations of the driver. As a consequence prototypes are used in the automobile industry within the design phases of new cars. This paper describes some examples about how the technology augmented reality can be used to support the automobile development process. In particular we use augmented reality for the completion of rudimentary prototypes. These prototypes consist of only a few real parts. Here the technology of augmented reality can be used for the completion of the prototype. The developed applications complete real automobile prototypes by virtual components to show design variants or to support design reviews. The evaluation of the applications was done at Volkswagen AG.

## 1 Introduction

Currently the worldwide market is defined by high dynamics, short innovation cycles and an increasing customer orientation [1]. In an intense degree this fact itself impinges on the automobile industry, where a rising competitive pressure between the car manufacturers as well as the corresponding suppliers exists. The number of automobile variants increase more and more, too, because single car modules are incrementally universal applicable and individually configurable [2]. At DaimlerChrysler 56 variants were added by introducing the new A-class - without considering extras like fittings or special configurations [3].

During the planning process of new cars, real prototypes are generally used e.g. to visualize design studies, assembly analyses or to support ergonomic decisions. The manufacturing of these prototypes is very cost intensive and takes generally a couple of up to several weeks. A subsequent modification of these real prototypes due to concept changes or undesirable developments is almost very cost intensive. For this reason real prototypes are only used in the later development phases, in which the car concept becomes more and more reliable. Since the last years VR-prototypes are increasingly used within the design stages of new cars. Digital prototypes based on 3D-CAD-models are presented on Power Walls or in CAVEs to generate a 3D-impression of the car. But the installation of Power Walls or CAVEs is very cost

intensive and requires a huge amount of space. Today, first a complete virtual prototype is created and only if every pre-analysis yields the desired results, a real prototype will be built.

In many cases, there exists no complete prototype of the car. Only some components like the platform (chassis) can be used. Other components like the car body or the interior design are only available as 3D-models in the computer. Here the technology of augmented reality can be used for the completion of these rudimentary prototypes. Thus using AR as a new man-machine interface, the planning and design process of new cars can be shortened and facilitated.

## 2 AR-Based Product Design in Automobile Industry

The method of completing imperfect automobile proto-types by virtual components using AR-technology is applied to the following field of applications at *Volkswagen AG*, department of commercial vehicles:

- Platform strategy
- Representation and verification of car components
- Test of car ergonomics in reality

This area of application assists and completes the today introduced development strategies and can be introduced into the product development process without any trouble.

### 2.1 Platform Strategy

Since several decades the global automobile industry is attempting to develop and produce world cars for the mass market which can be sold around the world with only minimal modifications. Bringing this strategy successfully to the mass market segment would result in tremendous economies of scale for the automotive industry and would drastically reduce development and manufacturing costs.

*Volkswagen* has cut production costs by building several models using common chassis and components as part of its so-called platform strategy [4]. This strategy is an essential success factor for *Volkswagen*. Using the platform strategy enormous savings has been achieved. In this context different car models use the same components and parts. A platform contains: front axle, steering and engine as well as stingers, basement, rear axle and the tank. As a result a new car is the combination of a given platform with an individual exterior and form. For example the *A4* platform is used for other models like the new *VW Golf*, *Audi A3* and *Audi TT*.

According to expert opinions the platform strategy enables enormous savings. At the purchase major numbers of the same parts can be ordered. This results in significant discounts. Less components and parts reduce the development and production times. 60 percent of the costs of a car are affected by the platform.

In the early design phases new car bodies can be virtually assembled on the given platform by using augmented reality (see figure 1). Therefore the physical car prototype is placed at a fixed position on a marker base to enable a tracking of the



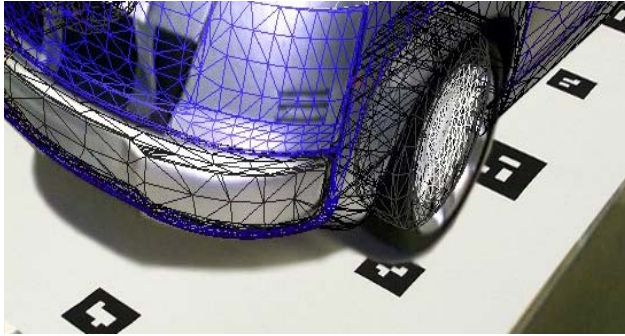
**Fig. 1.** Representation of a virtual front component of a VW Multivan on a real VW Microbus model

user's head position and an exact placement of the virtual car components. Various components like a new front, varying driving mirrors and buffer bars are augmented on the real car. They are generated from the company internal 3D-CAD-data. The usage of AR-technology in this field of application reduces the number of physical prototypes for a platform significantly, because car variants can be augmented on the real platform using existing company internal 3D-CAD-data. Quick changes of components can be done using only the 3D-CAD-system. The generation of the according virtual component for the AR-system requires less time and money than the creation of a new real physical prototype.

## 2.2 Representation and Verification of Car Components

A normal car consists of a high number of elements subsumed into components. Because of the integration of additional systems into the car e.g. to ensure safety (electronic break assistant, electronic stability program, etc.) or to increase the driving comfort (air suspension, dynamic multi-contour seat, automatic climate control, etc.) the number of car components increases more and more. Due to design decisions the available space for each component is limited. So one major point during the design of new cars is the maximal usage of available component space and the minimization of component distances [5].

It is always possible within the manufacturing process of single components, that tolerances exceed a specified maximum. This can result in problems during the packaging: components could not be arranged and assembled in the correct position and orientation or electromagnetic interferences between electronic systems or heating problems can occur because of the lowered distance between components. Today engineers use real models of the corresponding parts to accommodate their shapes in an iterative process.



**Fig. 2.** Superimposing of 3D-CAD-data on a real structure of a VW car model

The described problems during the manufacturing and packaging process can be reduced using the technology of AR. Therefore the real car is placed on a marker base (see previous chapter). Original 3D-CAD-data is augmented on the real structure of the car in a wire frame representation, as shown in figure 2. Within the superimposing of virtual components on the real car deviations respectively differences between real and virtual objects become obvious. Thus sources of errors can be identified quickly and the necessary modifications can be performed immediately. Thereby a specific and fast detection of the cause becomes possible.

### 2.3 Test of Car Ergonomics in Reality

In automobile industry a variety of methods are used especially for ergonomic analysis of the car interior. In this context the visibility of the surroundings from inside a car is one major topic [6].

Ergonomic simulation and analysis e.g. for reachability, accessibility of switches on the dash board, etc. are performed [7]. With these techniques, the reachability of the pedals, the steering wheel and the general seating position can be assessed – more difficult questions, such as the reachability of certain switches need to be answered, too. This described process is very time and cost intensive, because ergonomic analyses are usually done at real ergonomic prototypes. This ergonomic prototype is a model of the car interior, whose shape has nothing in common with the real car interior.

To reduce time and costs the technology of AR is used: The new interior is superimposed in a conventional car to enable the examination of the new interior under ergonomic aspects (see figure 3). Therefore a small marker base is mounted on the dash board for the tracking of the user's head position. As described before the data for the virtual car interior components is generated from company internal 3D-CAD-data. Now it is possible to analyze a variety of different (virtual) car interiors while sitting in a real car and getting a real haptic feedback. Even real test-drives are possible to get a closer-to-reality impression of the interior design and its ergonomic properties.



Fig. 3. Representation of a virtual car dashboard in a real VW Sharan

### 3 Realization of the Prototypes

#### 3.1 Hardware and Software

The developed AR-applications are running on a *Pentium IV* 1.5 GHz equipped with a *GeForce Quadro 4 980 XGL* graphics card. As output device the video-see-through HMD *VH-2002* from *Canon* is used to get a stereoscopic image of the AR-scene (see figure 4).

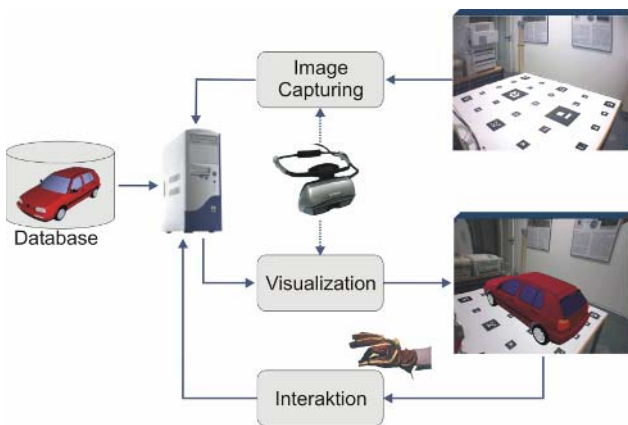


Fig. 4. Structure of the system



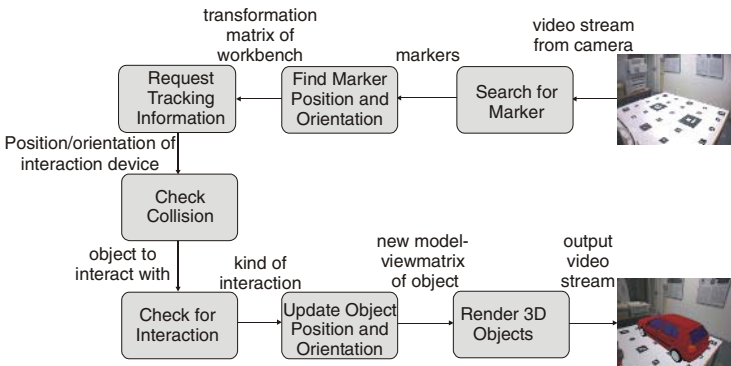
The tracking of the user's head position is done by an optical tracking method. Therefore the software *MuliMarker Tracking* from *HIT Lab* is used, which is based on the *ARToolKit* software [8, 9, 10]. The system offers the development of table-top AR-environments using markers for tracking. We use a base of one meter by one meter with different markers in different sizes. The markers reach from 4cm to 20cm, to track the user in close and in wide distance from the base. Thus the user has the possibility for an overview, to see the whole scene (e.g. the complete car in the scale 1:1) or to look into details (e.g. at the instrument panel). The user is able to interact with the virtual objects in the AR-scene by hand gestures. Therefore *Pinch Gloves* from *Virtual Technologies* are used. The sensors in each fingertip detect contacts between the fingers of each hand. This information is used for the gesture detection. The position of the hand is tracked electromagnetically by the *Fastrack* system from *Polhemus*.

### 3.2 Working Principle

The application's working principle can be divided roughly into three major steps:

1. Calculation of viewing direction
2. Analysis of user interaction
3. Rendering

In the first step two video images (right and left eye) are captured. Using these pictures and the visible markers the position and orientation of the user can be calculated. Subsequent the actual user interaction is analyzed. The last step is the computation of the position of each virtual object and it's rendering (for the right and left eye). The complete process is shown in figure 5.



**Fig. 5.** Function principle

For the detection of the markers and the computation of the user's viewing position as well as for the video handling the *ARToolKit*, version 3.15, is used.

For the interaction analysis first the hand position of the user must be detected. Therefore the tracking system *Fastrak* from *Polhemus* is used; it's electromagnetic sensor is mounted on a data glove worn by the user. The *Fastrak* system measures six

degrees of freedom (X, Y, Z, azimuth, elevation, roll). For the communication with the computer a standard RS-232 interface is used. The data is read in the ASCII format with a baud rate of 57600.

After the hand position detection the system analyzes the virtual objects the user wants to interact with. Therefore a collision detection between the virtual objects in the AR-scene and the position of the *Fastrak* sensor is implemented: Each object is assigned with a bounding volume and if the position of the *Fastrak* sensor is within a bounding volume, the object assigned to the bounding volume becomes the actual selection of the user.

In the next step the interaction done by the user is evaluated. The possible interactions depend on the result of the collision detection: If an object is selected, only the object-dependent actions are possible. These functions are:

- **Transform:** Transformation of an object or a group.
- **Rotate:** Rotation of an object or a group around it's center.
- **Scale:** Scale of an object or a group in x, y and z direction.
- **Delete:** Deletion of the selected object or group.
- **Lock/Unlock:** Locking/Unlocking of the actual selection for modifications.
- **Copy/Paste:** Cloning of an object or an group.
- **Select:** Add an object or a group to the actual selection.
- **Group/Ungroup:** Combining several objects or groups into an object group. Groups can be edit or delete during runtime.
- **Assembling:** Each object has one ore more connection points, which represent snap positions. To place an object at a specific position and orientation towards another object the user only selects the corresponding connection point of each object.

The interaction with the AR-scene is done by *Pinch Gloves*. Depending on the finger combination different interactions with the AR-scene are possible. When the condition of the *Pinch Glove* changes (contact areas are brought together or released) this information is transferred to the computer and stored in a terminal line buffer of the *Linux* kernel.

Beside the basic functions like *translate*, *rotate* or *scale* further context sensitive functions are available like *erase*, *copy* or *lock*. Furthermore additional management functions are available:

- **Load/Save:** The scene information is saved into a file containing the name, position and orientation of every object. To load a scene, the user selects the file in the listing dialogue.
- **Sensor Offset:** The *Fastrak* sensor shows a location dependant offset. To adjust this inaccuracy the user can define an offset to adjust the user's hand position relatively to the sensor.
- **Undo:** Canceling of the actual action.
- **Reset:** Initial state of the application.
- **Component Menu Load, Save, Add:** The current status of the component menu can be saved and loaded. Furthermore new virtual objects can be loaded into the component menu from a file or the virtual workbench (single objects or groups).

One major problem during the representation of the virtual car components on the workbench was the jittering. This reduced the quality of the whole application significantly and makes an interactive working with the AR-environment more and more difficult. For the reduction of the jittering and for the stabilization of the whole image a linear, time discrete and not recursive filter - a low pass-filter - is used [11].

The component menu, the yellow arrow representing the user's hand position and the other graphical illustration objects (e.g. coordinate system, bounding boxes, etc.) are implemented using *OpenGL* version 1.4 [12] and the *OpenGL Utility Toolkit* version 3.6. The component menu is created with basic *OpenGL* functions for drawing rectangular and triangular areas. To browse and interact with the components, a collision check is performed as described in chapter 3.1. The objects in the component menu are VRML-objects, too. The usage of these objects is similar to the usage of the objects available for the workbench.

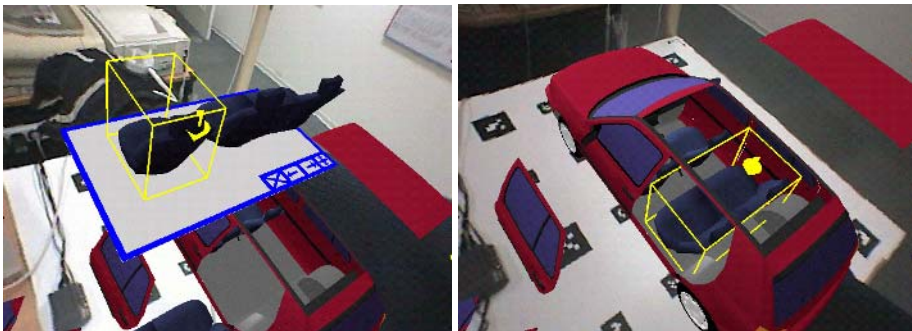
Having started the application each user equipped with an HMD can interact with the application using hand gestures. By finger tips he can open an context menu and a component menu. The component menu contains all the components witch can be added to the car, e.g. doors, mudguards, tires or seats. The user can select a component by his hand, grasp and put it on the virtual workbench (see figure 6a).

### 3.3 Working with the Application

He has additional buttons to *close the menu*, *go to the next item*, *go to the previous item* and to *move the menu*. If the user's hand touches an object of the menu or the workbench, a yellow bounding box will appear (see figure 6b).

By equipping each user with an HMD and a Pinch Glove a cooperative planning and design process is now practicable: each participant can interact with the virtual design and design changes and modifications become visible for everyone.

Additionally the planning stages can be saved, loaded and exported using the *VRML 2.0* standard. Those VRML-data may then be inserted into a VR scene to be displayed by a 3D-realtime rendering system like *RealiMation* or *DIVISION MockUp*.



**Fig. 6.** a) Placement of a component on the virtual workbench, b) Component menu containing car elements.

## 4 Usability and Performance Tests

The feedback of the trial testing of the first prototypes confirms the demand of new user friendly and easy to use interfaces. In several trials the involved persons needed only a short time of 5-10 minutes to understand the functions of the AR-based user interface. Unfortunately accurate placements, movements and alignments of the 3D-models in the AR-scene are difficult to realize. To overcome this a snap-function will be implemented. Using a marker based tracking method, it is necessary to warrant an adequate lighted up environment, whereas no mirroring or dazzling surface should appear. Otherwise a lost of tracking could accrue.

## 5 Summary and Outlook

In this paper we illustrated actual problems within the design processes of new cars in automobile industry. The creation of real prototypes is very time and cost intensive, whereas the usage of digital prototypes using *Power Walls* or *CAVEs* requires a huge amount of space and money, too. To overcome these problems the technology of AR is used. Here rudimentary car prototypes are completed by virtual components. Thus design decisions, packaging examinations and ergonomic analyses are supported using AR as a new man-machine interface.

The described prototypes have shown that an AR-based rapid prototyping of new concept cars is beneficial. The efficiency of the system mainly depends on the number and quality of construction elements available in the virtual component menu. The implementation of planning rules assists the user and prevents possible errors. We are currently refining the prototypes and preparing a concept car with a mobile AR-environment to enable a real test-drive. Thus the user can place virtual car components on a real car and make ergonomic evaluation of the car interior using AR-technology under real driving conditions.

## References

1. Knolmayer, G. F.: Kundenorientierung, Mass Customization und optimale Variantenvielfalt. In: Grünig, R., Pasquier, M.: Strategisches Management und Marketing, Bern-Stuttgart-Wien, 1999.
2. Martin, M. V. and Ishii, K.: Design for Variety: A Methodology for Understanding the Costs of Production Proliferation. In: Proceedings of the 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference, 1996, Irvine, California, 1996.
3. Kip: Testfahrt live im virtuellen Auto. In: VDI Nachrichten 12.06.1998.
4. Oehlschlaeger, H. and Balk, A.: Vorentwicklung in der Nutzfahrzeugindustrie. In: Tagungsband zur EUROFORM, Königswinter, 1999.
5. Gausemeier, J.; Ebbesmayer, P. and Kallmeyer, F.: Produktinnovation - Strategische Planung und Entwicklung der Produkte von morgen. Carl Hanser Verlag München Wien, 2001.
6. Jasnoch, U.; Anderson, B.; Koch, M. and Rix, J.: Beyond digital mock-ups: Human aspects in new products. In Proceedings of the Autofact 1997, Detroit, USA, 1997.

7. Rix, J.; Heidger, A.; Helmstädter, C.; Quester, R. and Ringhof, T.: Integration of the Virtual Human in the CA Design Review. In Landau, K.: *Man-Machine Interfaces*, 1999.
8. Kawashima, T.; Imamoto, K.; Kato, H.; Tachibana, K. and Billinghurst M.: Magic Paddle: A Tangible Augmented Reality Interface for Object Manipulation., In: *Proceedings of the Second International Symposium on Mixed Reality (ISMR) 2001*, Yokohama, Japan, 2001.
9. Billinghurst, M. and Kato, H.: Collaborative Mixed Reality. In: *Proceedings of the International Symposium on Mixed Reality (ISMR '99)*. *Mixed Reality-Merging Real and Virtual Worlds*, 1999.
10. Kato, H.; Billinghurst, M.; Poupyrev, I.; Imamoto, K.; Tachibana, K.: Virtual Object Manipulation on a Table-Top AR Environment. In: *Proceedings of IEEE and ACM International Symposium on Augmented Reality (ISAR) 2000*, Munich, Germany, 2000.
11. Gausemeier, J.; Matyszczok, C.; Radkowski, R.: Optical Tracking Stabilization using Low-Pass Filters. In: *The Second IEEE International Augmented Reality Toolkit Workshop*, Tokyo, Japan, 2003.
12. Segal, M. and Akeley, K.: *The OpenGL Graphics System: A Specification*, 1999.

# Real-Time Selective Scene Transfer

Min Tang<sup>1</sup>, Zheng-ming Ying<sup>1</sup>, Shang-ching Chou<sup>2</sup>, and Jin-xiang Dong<sup>1</sup>

<sup>1</sup> State Key Laboratory of CAD&CG, Zhejiang University,  
Hangzhou, 310027, China

{tang\_m, djsx}@zju.edu.cn, yingzm@163.net

<sup>2</sup> Department of Computer Science, Wichita State University,  
KS, 67220, U.S.A.

chou@cs.wichita.edu

**Abstract.** Applications like cooperative CAD and virtual touring on the Web need to transfer a scene consisting of models and textures from one computer to another. The scene may be very large, and the time to transfer the whole scene is significant. During transferring, the user has to wait until the whole scene is downloaded to the local machine. We propose a new approach to estimate the visual importance of each object or even of each polygon and a method that will selectively transfer the models and textures in the scene. In this way, we can reduce user-waiting time while keeping the image quality the user can see.

## 1 Introduction

With the development of CAD and network, people cooperate to work out on one scene through the Web. This brings the problem of transferring a scene from one computer (server) to another (client). However, it costs too much time on transferring a complicated scene with fine-detail models and high-resolution images so that the user has to wait for a long time to see the scene. Furthermore, if the scene is modified, and new models or textures are added later, the user has to wait again for downloading the new models and textures. This limits the cooperation in only simple scenes with coarse models and low-resolution images. Or the users have to cooperate on a high-speed network. Although the network speed is becoming higher and higher, the models and textures to transfer is becoming more and more complicated. We can divide the scene into several individual parts; the server will only transfer the part of models and textures that a client can see. The division can greatly help to reduce the waiting time. In this paper, we propose a method that enables to transfer a high quality scene on a low speed network with least a visual quality penalty. With a combination of portals or BSP trees techniques, our method can further reduce the waiting time.

Our method is based on the following observation. When previewing a scene, the user may not see all objects in the scene due to frustum culling or occlusion culling. Furthermore not all objects visible to the user can be seen clearly. Therefore we can use simpler objects and lower resolution textures for far objects. So we selectively transfer the models that can be seen and only transfer fine models for near objects, thus reducing the time to transfer the scene from the user's view. When the user later changes the point of view, new objects or better level of objects and textures will be

transferred. This kind of transfer-on-demand can significantly reduce the time to transfer a scene.

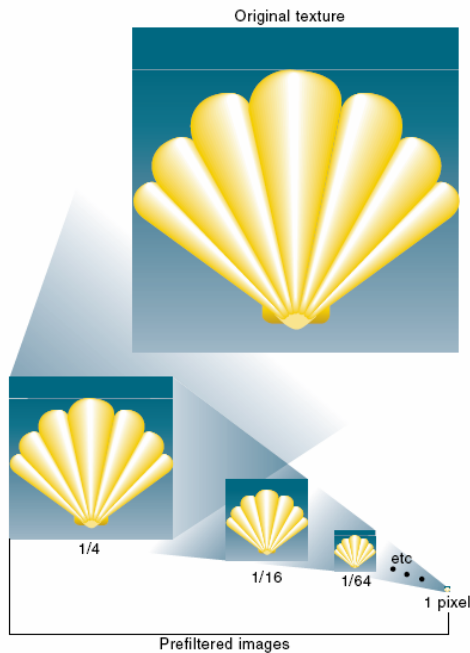
The method can be applied to model transferring and previewing on Web such as virtual touring, network games, or cooperation in the Web based CAD, or in any places where transferring scenes takes significant long time and the user only view from one direction at one time.

This paper is organized as follows. In Section 2, we summarize some previous work relevant to our method. In Section 3, we give a measurement to estimate the visual quality of a scene. In Section 4, we propose our scene transfer method according to this measurement. In Section 5, we give our results in some typical applications. Finally, we conclude the article and mention the possible future work.

## 2 Related Work

The most important techniques on which our method is based are LOD (Level Of Detail) of texture and multi-resolution meshes. LOD provides different levels of image quality for a texture, and multi-resolution meshes provide various levels of geometry.

In 1983[1] L. Williams proposed the technique of mip-map and introduced one texture with a texture array of different resolutions. The texture of lower resolution is generated from a texture of higher resolution. For near objects, the render engine can use high-resolution textures; for far objects, the engine can use low-resolution textures. Such kind of technique reduces the time to render and even results in a better



**Fig. 1.** Example for mip-maps



**Fig. 2.** Example for Progressive Meshes, the vertices numbers are 303, 203, 103, and 39 respectively

appearance for far objects. Obviously, low-resolution textures can reduce the transferring time through Web. Fig. 1 shows an example for mip-maps from [5].

Although LOD texture has been proposed very early, multi-resolution meshes [2][3] only become popular in recent years. Some recent graphic API like DirectX supports Progressive Mesh as one of its main features. Fig. 2 is a sample [6] for Progressive Meshes implemented with DirectX 9.0 [7].

We are not going to discuss these two techniques in detail. We just suppose the scene has been built with LOD textures and multi-resolution meshes, we only discuss how to select the appropriate level of texture or meshes to transfer.

There are already some commercial streaming methods for transferring sequential sound and video datum. They are apparently inappropriate for interactive 3D graphics applications. The work in [8] addresses the same problem; it gives a detailed analysis about accelerate techniques: acceleration of local rendering, benefit/cost optimization, graphics over network, and compression of 3D models. It also makes prediction of motions: standing, turning, moving. LOD and Impostor are used to simplify 3D scenes. Network bandwidth is taking into consideration in the cost computation. But since a lot of assumption was made, the author cannot give a practical application, and the acceleration on graphics hardware is not used.

The work of [10] addresses the problem of improving the MPEG compression of synthetic video sequences by exploiting the knowledge about the original 3D model. Paper [11] uses a real-time MPEG-4 streaming architecture to facilitate remote visualization of large scale 3D models on thin clients, the Graphic Processor Units (GPUs) are used to deal with most of the motion estimation process, which can be done in parallel to the encoding process. In both of them, the computation power of graphics hardware has not been fully exploited.

### 3 Quality Measurement

First, we need to determine how fine an object or texture should be. We call a model or a texture is in a full quality when it already achieves satisfactory visual quality, and no further transferring of finer levels is necessary.



To get objects and textures in a full quality, we must know how big the objects will be on the screen. For a level of a LOD texture, if one pixel in the texture covers no more than one pixel on the final image, we can say this detail level is good enough to produce a full quality. The multi-resolution meshes are in the same way, but it cannot be judged by the same “one-pixel” criteria as textures, the criteria for meshes are not that strict. Fortunately, we do not have to do everything ourselves, we can use a render engine that supports the multi-resolution mesh and LOD texture for our purpose. The maximal level of mesh or texture used by the render engine specifies the level we will need to create a full quality image.

We also need a value to indicate how much an object contributes to the full quality and need to know the visual quality at the client’s side.

At first glance, we can use the number of polygons in the scene to present the scene’s visual quality, because the more facets in a model, the more details it has. However, it cannot be used as the measurement of the scene quality, because the more facets it has, the larger its projection on the screen may be. We do not know whether it is a large scene with many simple, low quality models, or it is a small scene with a few high quality models. Our measurement uses a relative quality measurement instead of the absolute quality measurement. And we can see from following sections that the relative quality measurement can give us enough information about how important a model or texture is in visual contribution.

While the model at the client’s side may not reach the full quality, we can calculate the quality difference between them. From the above definition of full quality, we can find that there are two kinds of qualities: geometry quality and texture quality. Geometry quality indicates how fine our model is. If we have multi-resolution meshes with the number of meshes in the full-quality representation to be  $G_f$ , and the number of meshes of client’s mesh level to be  $G_c$ , then  $G_c/G_f$  is the geometry quality for this level of meshes. It is also possible to transfer only a part of a model. Suppose the number of meshes at the client is  $G_p$ , so the geometry quality  $G_j$  is  $G_j = (G_p / G_c) * (G_c / G_f) = G_p / G_f$ . Texture quality can be calculated the same way. Suppose  $T_{fx}$  and  $T_{fy}$  are the full-quality resolution of a texture, and  $T_{cx}$  and  $T_{cy}$  are the resolution for client’s detail level of the texture, then  $T_j = (T_{cx} * T_{cy}) / (T_{fx} * T_{fy})$  is the texture quality for the individual texture. We do not permit transferring only a part of a texture, because we cannot predict which part of the texture is needed later.

With the quality measurement for an individual model or texture, we can estimate the overall quality for the whole scene. The whole-scene quality value cannot help to select an individual object or texture. It can only provide an overview about how much the current image is from the full quality. On the server’s side, we can assign the whole scene with all models and textures at the full-quality level as quality value 1. If the other computer has only some parts of the scene, its quality value is between 0 and 1. The details are as follows.

We divide the quality of a scene into view-independent quality and view-dependent quality. So we can have up to four values to represent the scene quality. They are view-independent geometry quality, view-dependent geometry quality, view-independent texture quality and view-dependent texture quality. Let view-independent geometry quality for an object be  $G_j$ , then the overall view-independent geometry quality is defined as the average of all geometry qualities of all objects,

$$Q_{viG} = \sum_j G_j / N, \quad (1)$$

where  $N$  is the number of models.

The same equation applies to texture quality,

$$Q_{viT} = \sum_j T_j / M, \quad (2)$$

where  $M$  is the number of textures.

If the view-independent quality is 1 for both geometry and texture, we do not need to transfer models or textures anymore. We can say, the view-independent quality indicates the percentage that has been downloaded.

The view-dependent quality is related to view direction. It represents the image quality the user can see. Its value tells the user how much it differs from the full quality image. The quality value for an object is related with the percentage that covers the final viewing image. Suppose the final rendering image on the server is  $R_x * R_y$  resolution and the pixels for an object on the rendering image is  $R_o$ , then the important value for the object is  $R_o / (R_x * R_y)$ . The visual quality that the object contributes is then,

$$Q_{vdG,j} = (G_j R_j) / (R_x R_y). \quad (3)$$

Here  $G_j$  is view-dependent quality for an object. The  $G_f$  value of view-dependent is not the same as the  $G_f$  of view-independent. It also changes with user's view direction.

Obviously, for the objects that cannot be seen in current view, its quality value is zero.

The view-dependent texture quality value can be calculated in the same way,

$$Q_{vdT,j} = (T_j r_j) / (R_x R_y), \quad (4)$$

where  $r_j$  is the pixels covered by the texture.

When we decide which object or texture to transfer, we need a variable to indicate the visual quality increase if we transfer that object or texture. The variables can be easily get with

$$D_G = 1 - Q_{vdG,j}, \quad (5)$$

$$D_T = 1 - Q_{vdT,j}.$$

When we want to transfer the objects and textures, we may have to choose between transferring objects or textures. However, it is dependent on the user's need, especially on the importance of texture in the application.  $K * D_G$  will increase geometry importance to texture importance if  $k$  is greater than 1, and will decrease it if  $k$  is less than 1.

## 4 Model Transfer Scheme

Now we propose a model transfer scheme according to the measurement in the preceding section. The scheme we use greatly increases the visual quality at the client's side. The following are the techniques we use to achieve the goal.

### 4.1 Selective Model Transfer

With the quality value calculated using the method in last section, we can choose the objects or textures that has the largest  $D_G$  (or  $K$ -modulated  $D_G$ ) or  $D_T$ . However, to get the quality for each objects and textures, we need to render the scene from the user's point of view on the server to do frustum culling and occlusion culling, and we can get the pixels each object covers in the image. This may not be a problem on today's hardware. Even if we do not use LOD textures and multi-resolution meshes, the culling can give us much more information about which objects to transfer.

We can see from the definition of the four qualities that view-independent quality is irrelevant to the server side, it is only meaningful for the client side in measuring how much of the scene are downloaded. So we will only need to calculate the view-dependent quality at the server side.

The process of selective model transfer is as follows:

- (1) Rendering the scene at the server side.
- (2) Obtain the pixel coverage for each object and texture, multi-resolution mesh level for each multi-resolution mesh and LOD texture level for each LOD texture. The pixel coverage can be implemented using OpenGL [4], while the LOD texture level can be obtained with OpenGL mip-map texture routines.
- (3) Use the equations discussed in last section to calculate the visual importance of each object and texture.
- (4) Sort the objects and textures by importance value, and transfer them in that order.

### 4.2 Clip and Combine

Although we can only selectively transfer the objects, we may still suffer a long waiting problem if the single object is too complicated and need too much time to transfer over a network. For polygon objects, we can transfer only one part of the object at one time, then transfer other parts later. We will only need to add a flag indicating whether the individual polygon has been transferred or not. And it is also easy to get whether the individual polygon can be seen from current point of view.

### 4.3 Predict Incoming Models and Textures

The view-dependent quality 1 means we need not transfer any finer models or textures to increase the image quality at the client side. But the user's view may change in a short time, and we want to pre-transfer some models or textures beforehand. If we really need the models or textures later, then we do not need to transfer them and there is no waiting time from transferring the models and textures.

In different applications, different predicting scheme may be used. We will not propose an algorithm of predicting the incoming models and textures. Instead we just indicate two typical circumstances. In virtual touring applications, pre-transferring models and textures along the view direction is preferred. In cooperative CAD system, models viewed may have more priority. However, it depends on the user's way of changing views.

## 5 Implementation and Examples

Before the server decides what to transfer to the client, it must render the scene from the user's point of view to get the level of multi-resolution meshes and LOD texture. Multi-resolution meshes are not implemented in our system because OpenGL does not directly support it yet. Since we use OpenGL to do the work, and OpenGL is hardware-supported on most popular video cards, its time can be ignored comparing with the transfer time on Web.

To get the pixel coverage for each object and the pixel coverage for each texture, and to determine whether a polygon can be seen, we use different encoding method in different situations.

A straightforward way is to represent the RGB values as a 24-bit integer. We render the scene assigning the object id as the color of the object, then render the scene and read the color buffer. We can get the pixel coverage for objects by checking the contents of the color buffer.

The polygon culling information can be obtained by the same way except it uses the face id instead of object id.

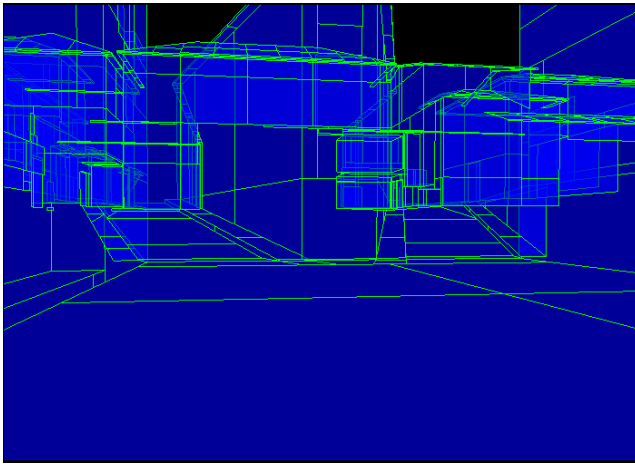
Since the LOD texture level can be gained by OpenGL's mip-map mechanism, we can build a mip-map texture that has different values for different levels of LOD. For example, the values can be equal to the number of levels. That is, if we have a LOD



**Fig. 3.** A Scene contains 856 facets and 36 textures

texture that has 4 different levels, we can build the mip-map texture in OpenGL. Suppose we use the lower 2 bits to represent the level of detail, and the higher 22 bits the texture id. So the color of the first level is  $A$ , the color of the second level is  $A+1$ , the third is  $A+2$ , the fourth is  $A+3$ . Then we render the scene from client's point of view; OpenGL will select the appropriate level of detail for us. We will only need to read back the color buffer in RGB format, and extract the first 22 bits to get the texture id and the lower 2 bits to get the level detail for the texture.

If we do not have much textures and models, we can merge the above three rendering passes into two or one pass using a bit different encoding way.



**Fig. 4.** Rendering in Wire-frame mode for Fig. 3



**Fig. 5.** A Scene contains 573 facets and 29 textures

The following are some examples for data transfer from the server to its game clients. Our implementation is based on the open source WinBsp from [11]. It loads wad files and builds a BSP tree for the scene. The whole scene contains 8528 facets and 89 different textures. By first clipping using BSP tree, only 856 facets and 36 textures need to be transferred for the scene in Fig. 3. And by the LOD properties of textures and applying our method, among the 36 textures, only 11 are in level A, 17 in level A+1, and 8 in level A+2. So the ratio reduced can be estimated by  $(11+17/4+8/16)/36 = 43.75\%$ . Fig. 4 shows the scene in the debug mode, and all the facets are rendered in wire-frame mode. For the scene in Fig. 5, there are totally 573 facets and 29 textures. For all the textures, 8 in level A, 10 in level A+1, and 11 in level A+2. So the ratio will be  $(8+10/4+11/16)/29 = 38.58\%$ .

Although our demo system is implemented using OpenGL, our method do not limited to a special Graphic API(OpenGL/DirectX), it is a general method. For example, in a Web environment, it can be implemented using Java/Java3D.

## 6 Conclusion and Future Work

The method has its shortcomings. It ignores the effects of reflection or refraction, or the global illumination. And it only considers the objects and textures that can be seen in the point of view. The objects that cast shadows in the current point of view may have zero or low visual importance, and might be never transferred to the client's side. Furthermore the server has to be told the view of the client. In spite of the above shortcomings, it can still transfer scenes of significantly good quality within short time.

In addition of overcoming above shortcomings, future work may include algorithms on predicting the incoming models and textures. Texture segment transferring that transfers only parts of a texture is also a possible future work.

Acknowledgment: we thank the anonymous author of WinBsp who kindly provides the source and data file for our implementation.

## References

1. Williams, L.: Pyramidal Parametrics. Proceedings of SIGGRAPH83 (1983) 1-11
2. Eck, M., DeRose, T., *et al.*: Multiresolution Analysis of Arbitrary Meshes, Proceedings of SIGGRAPH95(1995) 173-182
3. Hoppe, H.: Progressive Meshes, Proceedings of SIGGRAPH96 (1996) 99-108
4. Segal, M. and Akeley, K.: The OpenGL Graphics System: A Specification (Version 1.2), 1998.
5. Shreiner, D., Woo, M., *et al.*: OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.4, 4th Edition, Addison-Wesley Pub Co, November 14, 2003.
6. Luna, F.: Introduction to 3D GAME Programming with DirectX 9.0, Wordware Publishing, (2003)
7. Microsoft Corp.: DirectX 9.0 SDK Update (Summer 2003) C++ Documentation, [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/directx9\\_c/directx9\\_cpp.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/directx9_c/directx9_cpp.asp), (2003)
8. Teler, E., Lischinski, D.: Streaming of Complex 3D Scenes for Remote Walkthroughs, Proceedings of EUROGRAPHICS 2001, Manchester, UK, September (2001)

9. Quaglia, D. and Gattuso, A.: Model-Based {MPEG} Compression of Synthetic Video Sequences, Proceedings of IEEE Int. Conf. on Image Processing, Singapore, Oct. (2004) 1109--1112
10. Liang, C., Anusheel, B., *et al*: Real-Time 3D Graphics Streaming using MPEG-4, Proceedings of BroadWise'04, San Jose, CA, USA, (2004)
11. Great3D, WinBsp sources, <http://www.190hz.com/index.htm>, (2003)

# Design and Implementation of a Collaborative Virtual Shopping System

Lu Ye<sup>1,2</sup>, Bing Xu<sup>2</sup>, Qingge Ji<sup>3</sup>, Zhigeng Pan<sup>2</sup>, and Hongwei Yang<sup>2</sup>

<sup>1</sup> Department of Computer and Electronic Engineering,  
Zhejiang University of Science and Technology, Hangzhou 310012, P.R. China  
yeluee@yahoo.com.cn

<sup>2</sup> College of Computer Science, Zhejiang University,  
310027, Hangzhou, P.R. China

<sup>3</sup> Department of Computer Science, Sun Yat-Sen University,  
510275, Guangzhou, P.R. China  
{xubin, jqg, zgpan, yanghongwei}@cad.zju.edu.cn

**Abstract.** E-commerce has rapidly grown with the advent of information and communication technologies. It has also become a promising field for applying VR and AI techniques. However, customers are not provided with the realistic shopping experience as they will enjoy in an actual store or mall. Therefore this paper proposes a multi-agent support for collaborative shopping systems by focusing on simulation and interaction. With combination of the sociality with a virtual environment, the proposed Easy Mall system cannot only imitate real interaction among target customers who are favorable to the same products, but also support the communication between multiple avatars. The system is implemented using VRML, intelligent agents and computer network technologies.

## 1 Introduction

With the rapid expansion of the Internet, E-commerce becomes more popular. Nevertheless, existing E-commerce applications on the Web provide users a relatively simple and browser-based interface to access available products. The customers are mainly kept separated as if he/she is in an empty shop. Thus, customers are not provided with the same shopping experience as they would enjoy in an actual store or mall [1].

In collaborative shopping, people can go to the virtual shopping mall along with relatives or friends. Multiple customers can join the collaborative session to communicate with each other. In our paper the EasyMall system is presented to simulate the actual shopping experience through implementing the collaborative shopping based on creation of the multi-agent model in a virtual shopping mall environment. Multiple agents can search and recommend products according to the customer's preference. Using an avatar chosen from a wide range of avatar identities, the customers can walk around the virtual environment, look over and manipulate the products that he is interested in, and order goods through a secure transaction system.

Our collaborative shopping system also creates the same virtual situation as in the real world: multiple customers can join and do the shopping together if it is found that they are looking for the products in the same products area. They can chat with each other, ask others' suggestions and find the desired products more efficiently.



The remainder of this paper is organized as follows: after reviewing the previous work related to our research in Section 2, we give an overview of the system in Section 3. In Section 4 we describe the fundamental technology in our collaborative shopping procedure. Implementation of collaborative shopping in EasyMall is presented in Section 5. Finally, we summarize the contributions with future research directions in Section 6.

## 2 Related Work

There has been a tremendous amount of previous work on creating a collaborative environment and interaction spaces across networks, with notable contributions from the fields of Computer Supported Collaborative Work, Groupware, and Computer Human Interaction [2]. In this section we briefly review the previous works that are related to our work.

Shen [1] presented vCOM, a VRML and Java3D-based prototype, which permits users to navigate around virtual e-commerce worlds and perform collaborative shopping and intelligent searches with the assistance of software agents. Real-time interactions between the entities in this shared environment are implemented over the High Level Architecture (HLA), an IEEE standard for distributed simulations and modeling.

Puglia [2] proposed a component-based architecture for collaboration that provides shared navigation of the WWW along with an EJB-based server implementation. As a particular application built on this architecture, they present Multi E-Commerce, through which multiple users can participate in virtual shopping trips among multiple shopping sites. Promondia [3] provided a client-server architecture, based on the distributed of applets, to support collaborative tasks such as text-based chat, shared whiteboards, voting and surveys, and implements a system where the interaction is limited to one Promondia server distributing page and services.

Our EasyMall system is a multi-server based on intelligent software agents [4], which uses blaxxun to implement the interaction between customers and the virtual environment. Its goal is to create an interactive virtual mall with integrated E-commerce, agent technology and virtual reality. VRML and Java3D technology are employed to provide an avatar for each customer who can communicate with other avatars as well as products in a shopping mall and agents [5].

## 3 System Overview

EasyMall system is based on the client-server architecture which supports collaborative shopping. Basically, the collaborative shopping system can be divided into four layers as shown in Figure 1.

**Client Layer:** As for the distribution of the shopping, every user in the client layer may collect items from the different merchant sites he is interested in. The locations the user is visiting are expressed in terms of the URLs of the 3D virtual environment he currently navigates from his terminal.

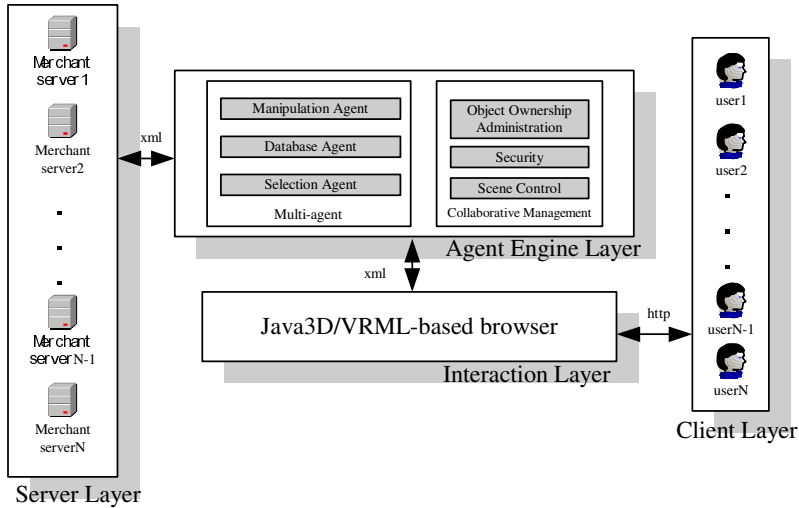


Fig. 1. An overview of the EasyMall system architecture

**Interaction Layer:** At client site, the interaction layer provides the interaction interface for every user. The browser based on Java3D or VRML mode offers some performance advantages to users, such as choosing the avatar, chatting between avatars and controlling the roam of avatar, etc.

**Agent Engine Layer:** The agent engine layer plays a critical role, which supports the communication of the collaborative shopping in virtual environments. The agent engine layer contains two consecutive high-level processes: multi-agent and collaborative management. In the multi-agent level, there are three agents: recommendation agent, database agent, and manipulation agent. A detailed description of these agents will be presented in Section 4.1. The collaborative management level is also included in the layer, which is composed of object ownership administrator, security and scene control. In Section 4.2, we will discuss these in detail.

**Server Layer:** The server layer includes multiple merchant servers. Every merchant server stores the various products and merchants information. The agent engine layer communicates with the server layer in XML and accesses the related data.

## 4 Agent-Based Collaborative Shopping

In this section, we introduce the core module of the system. We start with explaining the main components of the intelligent software agents. Then, we describe the collaborative management and its components.

### 4.1 Multi-agent Mechanism

To offer accurate and required products that the customer wants to purchase, the EasyMall system uses intelligent software agents, which can provide the selection and

manipulation of the products to customer according to customers’ preference. During the procedure, database agent is also used to manage and manipulate varieties of data such as products database, sales database and customer profile data. The multi-agent framework is showed in Figure 2.

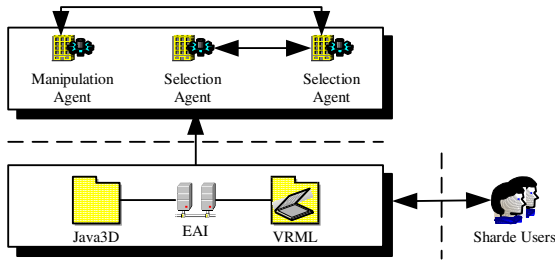


Fig. 2. Multi-agent structure

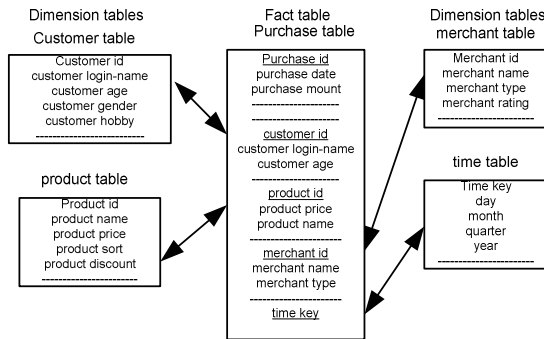


Fig. 3. Star schema of a data warehouse for purchase

In the following section we describe how the agents guide customers to purchase.

### 4.1.1 Database Agent

In current E-commerce, the most popular data model for a data warehouse is a multi-dimensional model. Such a model can exist in the form of a star schema, a snowflake schema, or a fact constellation schema.

Figure 3 shows the star schema which is used in the EasyMall system. Purchasing is considered from four dimensions, namely, products, time, customer, and merchant. The schema has a central fact table for purchase which contains keys to each of the four dimensions. To minimize the size of the fact table, dimension identifiers (such as time\_key and merchant\_id) are system-generated identifiers [7].

In the star schema, only one table represents each dimension, and each table contains a set of attributes. For example, the customer dimension table contains the

attribute set {id, name, age, gender and hobby}. In order to divide customers into different groups and recommend products to target group, we design database agent based on the star schema so that it can satisfy the need to understand the behavior of business units such as customers and products.

Database agent stores all information concerning the customer profile, the purchase history records, virtual environment including characteristic of avatars, the history of their encounters with other avatars and the clicks of customer, so that the customers can not only be recommended the favorable products but also be presented with information about the current situation and encounters made by the avatar in the virtual mall when the customer logs in and chooses his/her avatar.

#### 4.1.2 Product Selection Agent

When the customers login the EasyMall, they get across overwhelming information and look for help in making selection from many products. In dealing with this problem, the product selection agent is built to retrieve product information that really interests the customers. Two kinds of products recommending methods are adopted in the product selection process. For products that the customers often purchase, there exist a lot of customer profiles such as the previous buying behavior, the browsing history, and personal information about a particular customer. At the same time, experiential knowledge about customers plays an important role in the product selection agent since the customer has his specialized knowledge on the products and thus gives some appropriate specifications. According to the characteristic of this type of product, the module employs an interactive means to evaluate and recommend such products.

On the one hand, the interactive interface is developed to offer guidance by enquiring goal-directed questions and present product alternatives to help the customer decide [11]. Here, the features in the interactive interface are the keywords associated with a product, available from the on-line database. In the interactive mode, customers can evaluate degree of interest for each feature from his experiential knowledge. If the customers are not satisfied with the search products, the agent can also offer the customers to modify the search result by modifying some features until they find the optimal products.

On the other hand, customer information can be obtained by monitoring the customer activities during the navigation of a certain product area and by recording the contents the customer has read. Using the customer information, the statistic methods can further analyze the importance degree of each feature' value for the same kind of products according to different customers' preference in order to conclude what products meet the customer's requirements.

Unlike the above method, for the products such as TVs or mobile phones that a typical consumer does not purchase often, there may not be enough information to be analyzed. To deal with this type of products, the agent requires the customers to manually rate a few products according to his preference. Once the rated products have been collected, the combination of the genetic algorithm (GA) with  $k$  nearest neighbor ( $k - NN$ ) technologies is adopted to match customer interests.

### 4.1.3 Product Manipulation Agent

As to the manipulation agent, it allows customer to control an avatar to do some motions so as to inspect the products he/she chooses through the animation playing in a third person view point or operate products with the first person viewpoint. Product manipulation agent is also used to control the behavior of objects. When the avatar picks up a product, he can look over the product from different angles and can also read the relevant information about the product through text, image and video.

For example, when an avatar enters the clothing zone, she can select the favorable product and try it on. If she is not satisfied with the color or texture of the skirt, the product manipulation agent can help her to change the texture based on the texture morphing method [8] (depicted on Figure 4).



**Fig. 4.** Simulating results of a skirt

## 4.2 Collaborative Management

The collaborative management is the central of the collaborative shopping system. The current virtual market places often lack the emulation of the social interaction factors [6]. So our system combines the sociality with virtual environments. The customer chooses his/her avatar representative by registering information, and changes the role from “hollow-man” to human. The customers may “walk” and select products in the virtual environment by avatar, they can also invite other avatars or accept invitations from other avatars.

The collaborative management allows the customers to manipulate the products in a natural manner such as a customer selects T-shirt from the costume area. In the following section, we will introduce the components of collaborative management.

### 4.2.1 Object Ownership Administration

In the procedure of collaborative shopping, the first thing we need do is the ownership of object. The collaborative management module provides the ownership core based on mutex lock mechanism for exchanging attribute ownership among multiple avatars in the distributed virtual space so as to ensure that only one avatar at a time has access to the product. In the following, the Pseudo code of dealing with the mutex lock is given:

```

pthread_a mutex; //definition of the mutex semaphore

pthread_mutex_init (mutex, mutexattr); //initialize the mutex variable,
the first parameter is the variable that will be initialized; the second
parameter is the attribute of the mutex variable.

pthread_mutex_lock (mutex); //lock the mutex variable

pthread_mutex_unlock (mutex); //unlock the mutex variable

```

The mutex lock mechanism can only allow one avatar to handle the object, once the object has its owner; other avatars exclusively manipulate the attributes of an owned object (such as orientation and location). If another avatar attempts to operate the object, he is added in the waiting queue until the ownership of object is released. For instance, when an avatar clicks the T-shirt that he wants to operate, he/she first needs to request the ownership of object by using the ownership core. If the ownership of the T-shirt has already been distributed, the avatar has to wait until the ownership is given back. Then, ownership core assigns the ownership to the avatar that is in the waiting queue according to the priority and waiting time. After the avatar puts the T-shirt back, the module releases the ownership to its original owner.

#### 4.2.2 Security

The security, another crucial model, is responsible for controlling the privacy of customers and merchants. In fact, the collaborative shopping in virtual environments is done through the interaction of remote objects between the client layer and server layer. In the system, we use Java and Java3D to implement the object remote control. The problem we encounter is due to restrictions imposed by the Java security model on remote access to an applet. Firstly, the absence of support for multiple inheritances in Java makes it impossible to have an applet, which already inherits from the Java Applet class and extends the Unicast Remote Object (which is necessary to make an object remote). Secondly, preventing the server layer from opening a separate socket to actively “write” on a remote object bound to the applet is another server security obstacle. The problem is that an applet may not open a server socket to listen to requests from the arbitrary hosts [2].

As to privacy, we use the security model to allow the mechanism to degrade or enhance the customers and merchants’ privacy. On the one hand, the security model enables the possibility to provide anonymity to customers since it can prevent cookies and other private data from being transmitted to the merchants, acting as a filter between the two. On the other hand, the information that is available to the security model related to customer identity, preferences and shopping habits will be of great value to merchants and vendors.

#### 4.2.3 Scene Control

Traditional distributed systems are based on a central server model. In this model clients communicate with a central server, which manages the entire system and informs clients of any updates and changes in the state of avatars and objects. Clients only communicate with the stand-alone server, which contains the entire scene database and tracks all objects of interest within the system [9]. Especially, in large virtual worlds the number of objects that require certain synchronization or update messages to be transferred over a network can slow down the interaction of the individual user

with the shared world in an unreasonable way [2]. A solution for this problem is the subdivision of large virtual worlds into several regions or zones [10].

For the collaborative shopping environment, we partition the virtual environment into several separate areas. Each of the areas in the virtual environment has its own boundary and world coordination. In our approach, we use the Proximity Sensor of VRML to track the avatar's coordination. The coordination is represented by  $[Q, X, Y, Z]$ , where  $Q$  represents the id of certain shopping area which the avatar is in,  $(X, Y, Z)$  represents the avatar's position information in the shopping area. For example, when an avatar walks inside the boundary of T-shirt area, the sensor transmits the information to scene control model, so it refreshes and synchronously updates the corresponding data on the participating avatar.

### 5 Implementation of Collaborative Shopping

We implemented a multi-user collaborative shopping virtual environment by using VRML, Java3D, intelligent agents and computer network technology. In our system, we provide the user with simple and efficient navigation tools to visit the multiple virtual malls (depicted in Figure 5). The EasyMall system provides the multi-agent model to allow the avatar to communicate with the intelligent guider. When customers enter the virtual mall, the intelligent guider helps customers find out what they really want. The customers can simply identify the type of products they need by describing the features or specific functions of the products.

Customers may interact with other avatars in the shopping procedure. If two avatars meet in a T-shirt store, they will start a synchronous communication with each other by the text-based chatting and discuss their current hobby or exchanging their advice on the same T-shirt. Besides chatting, the customers can join and do the shopping together, as it is done in the real life. If they have noticed that they are looking

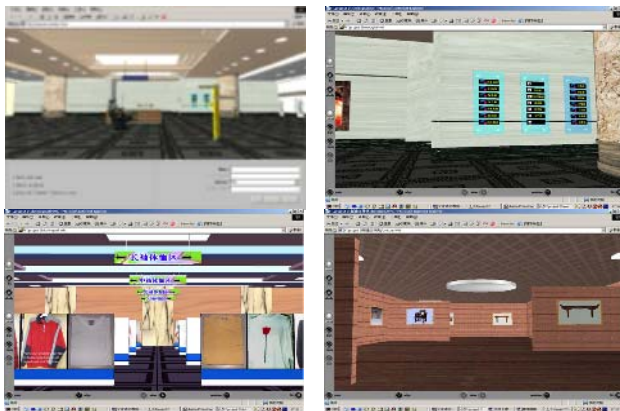


Fig. 5. Multiple virtual malls (Entrance, Products Selection Area, Garment Area and Furniture Area)



Fig. 6. Collaborative shopping in EasyMall

for goods within the same category, they could talk to each other, ask the others' opinions which will more effectively help the customers find and choose the desired goods [2]. Figure 6 shows an example in the EasyMall system where two avatars collaboratively buy a T-shirt. The female avatar finds a beautiful T-shirt, shows it to the male avatar and asks his opinion.

## 6 Conclusion and Future Work

We have presented EasyMall, a system aimed at supporting collaborative shopping based on intelligent agents in virtual environments. With the creation and application of the 3D virtual shopping mall, the customers may simulate the shopping experience in a real world by avatars. We have integrated virtual reality and intelligent systems technologies into E-commerce to generate a collaborative shopping prototype system. The prototype not only imitates realistic communication among special customers who are favorable to the same products, but also provides human-like interactive interfaces.

Our future work is to design a virtual memory with the smart object technology to implement multiple avatars' autonomic behaviors in virtual environments, which can improve the performance of collaborative shopping. In addition, we will concentrate on improving the efficiency of recommendation and the avatars with affective expression.

## Acknowledgements

This project is jointly supported by 973 High Tech Program of China (Grant no : 2002CB312100), European ELVIS Project, National Natural Science Foundation



(Grant No. 60473109), and Excellent Young Teacher Award Project of the Ministry of Education of China.

## References

1. Shen, X., Radakrishnan, T., Georganas, N.D.: vCOM: Electronic commerce in a collaborative virtual world. *Electronic Commerce Research and Applications*, 1 (2002) 281-300
2. Puglia, S., Carter, R. and Jain, R.: MultECommerce: A distributed architecture for collaborative shopping on the WWW. *ACM Conference on Electronic Commerce*, (2000) 215-224
3. Gall, U., Hauck, F.J.: Promondia: A Java-Based Framework for Real-time Group Communication in the Web. *Proceedings of the Sixth International World Wide Web Conference Santa Clara, California, USA*, (1997) 917-926
4. Xu, B., Pan, Z.G., Yang, H.W.: Agent-based Model for Intelligent Shopping Assistant and its Application. *The first Conference on Affective Computing and Intelligent Interaction*, Beijing, (2003) 306-311
5. Ji, Q., Hong, B., Wang, D.: Mobile Agent based Prototype of Heterogeneous Distributed Virtual Environment Systems. *Journal of Systems Engineering and Electronics*, 11(2) (2000) 61-65
6. Schuemmer, T.: GAMA-Mall-Shopping in Communities. *Second International Workshop on Electronic Commerce (WELCOM'01)*, Heidelberg, (2001) 51-62
7. Han, Jiawei, Kamber, Micheline: *Data Mining Concepts and Techniques*. Morgan Kaufmann Publishers (2000)
8. Wang, P., Zhang, M.M., Pan, Z.G.: New texture morphing method for visual presentation of textile product. *Second Intl. Conf. on Image and Graphics*, HeFei, China, (2002) 1011-1016
9. Morillo, P., Fernandez, M., Pelechano, N.: A Grid Representation for Distributed Virtual Environments. *2003 Annual Crossgrid Project Workshop & 1st European Across Grids Conference*, (2003) 182-189
10. Broll, W. *Populating the Internet: Supporting Multiple Users and Shared Applications with VRML*. *Proceeding of the VRML'97 Symposium*, Monterey, CA, ACM, (1997) 87-94
11. Lee, W., Liu, C., Lu, C.: Intelligent agent-based systems for personalized recommendations in Internet commerce. *Expert Systems with Applications*, 22 (2002) 275-284

# Digital Virtual Human Based Distance Education System

Liyan Liu<sup>1,2</sup>, Shaorong Wang<sup>1,2</sup>, Fucang Jia<sup>1,2</sup>, Hua Li<sup>1</sup>, and Zongkai Lin<sup>1</sup>

<sup>1</sup> Key Laboratory of Intelligent Information Processing, Institute of Computing Technology,  
Chinese Academy of Sciences, Beijing 100080, China

<sup>2</sup> Graduate School of the Chinese Academy of Sciences, Beijing 100039, China  
{lyliu, shrwang, fcjia, lihua, lzk}@ict.ac.cn

**Abstract.** With Digital Virtual Human as a foundation, a distance education prototype system is presented in the paper. The implementation methods of two key technologies in the system are proposed and described. One is to manage Digital Virtual Human mass data based on data grid technology, and the other is to support group collaborations with application collaboration tools. A case study is used to show the validity and feasibility of the system.

## 1 Introduction

Since the appearance of Computer Supported Cooperative Work (CSCW) it has been applied to many application areas, among which distance education is a typical one. In this paper, with Digital Virtual Human as a foundation, a distance education prototype system has been built in local area network.

Digital Virtual Human stands for the newest achievement of the combination of medicine and computer science. The total amount of its datasets is already measured in terabytes [1,2]. Furthermore the researchers that need to access and analyze these data are almost geographically distributed. Both of them result in complex and stringent performance demands that are not satisfied by any existing data management infrastructure. Data grid technology enables coordinated sharing of heterogeneous distributed storage resources and digital entities based on local and global policies across administrative domains in a virtual space [4-7]. Having been witnessed a booming development since its emergence, data grid has gradually become a new way to manage mass data.

Another crux of establishing such a system is application collaboration to support direct interoperation among group members, because the segmentation and registration of Digital Virtual Human images have not yet been accomplished completely automatically. Existing collaboration tools like whiteboard support simple interactions among members through some multimedia means, but it is helpless in modifying on operating results. Application sharing tools excel the whiteboard by covering this function, but only one member is allowed to perform operations at a moment. Direct interoperations, especially comprehensive group collaborations cannot be obtained. So in the proposed system, our previously developed software for image segmentation and registration is upgraded to redirect to be an application collaboration tool. Next we will give our system a detailed description.

## 2 System Functions and Structure

### 2.1 System Functions

Fig. 1 shows the main interface of the system, which is composed of tools area, text area, graphics area and student list. The system provides rich tools including mass data management, graphics tools and speech tools etc, realizes functionalities such as synchronous or asynchronous teaching, discussing, checking and tutoring. In the “text area”, a group members list is displayed which supports communication among members; images and operating results of teachers or students are shown in the “graphics area”; as to “student list”, it lists all on-line students, through which teachers may learn current students’ status. For instance, the student ‘John’ shown in red (Fig. 1) indicates that he is communicating with his teacher now. In Section 4, coupled with an example the system functions will be introduced in more detail.

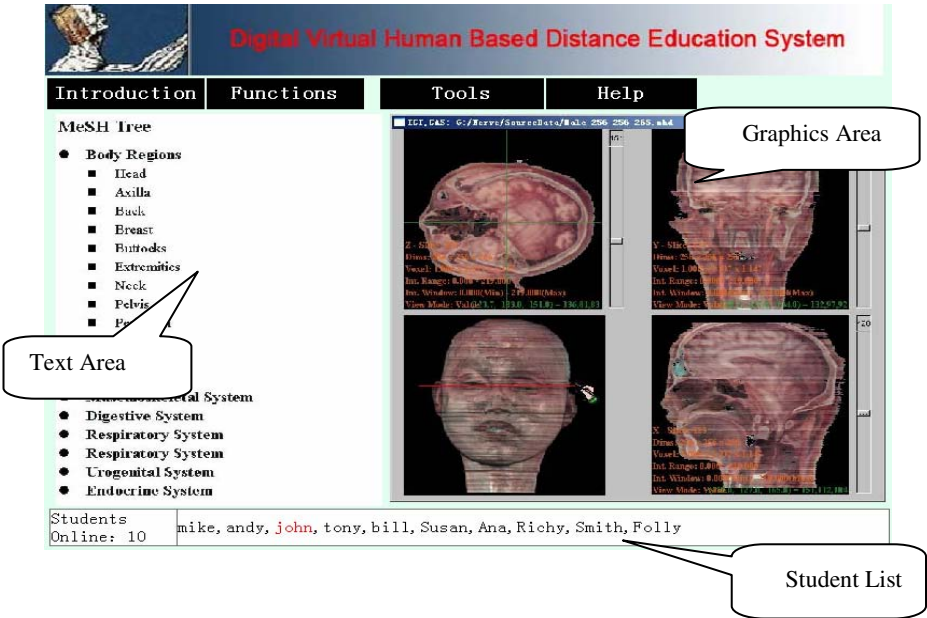
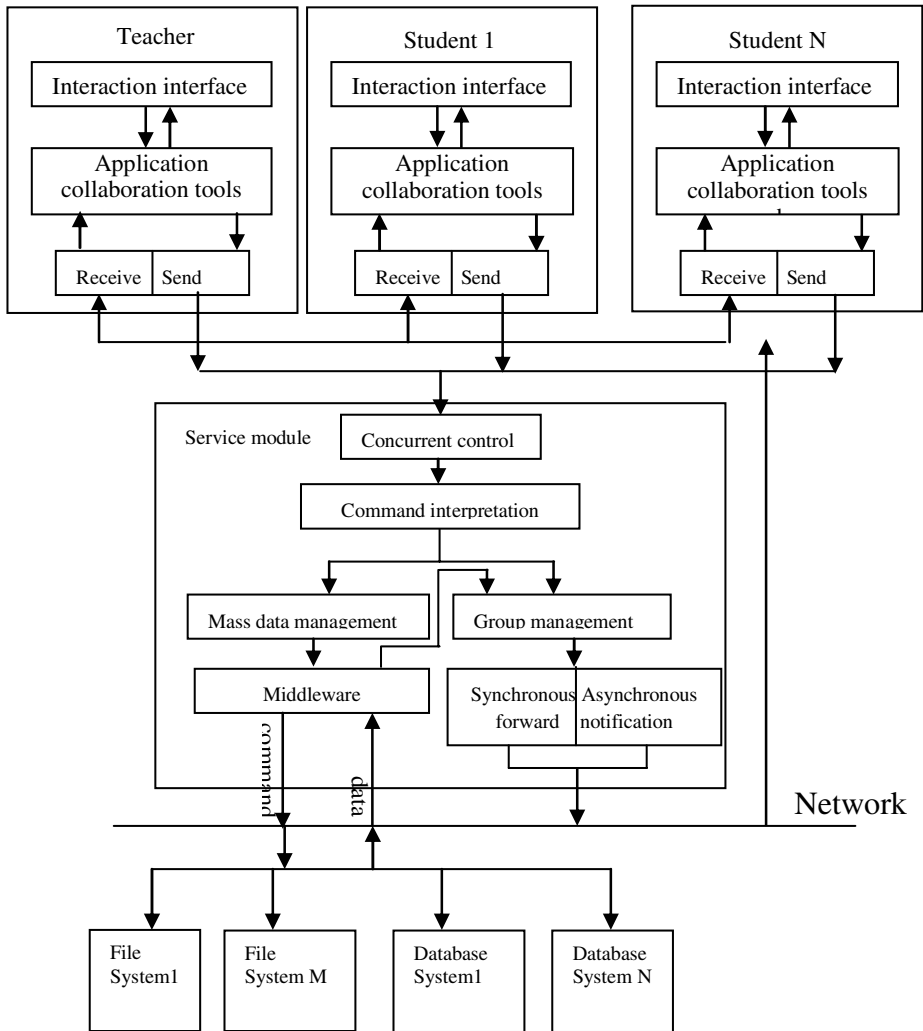


Fig. 1. Interface of our prototype system

### 2.2 System Structure

Our prototype system contains teacher module, student module, service module, and storage system, as shown in detail in the following. See also Fig. 2.

- Through interaction interface, teachers or students use application collaboration tools, and their cooperation commands are transferred to the service module;
- Service module analyzes and processes received commands:



**Fig. 2.** The structure of the prototype system

- “Concurrent Control” is responsible for lining commands to ensure an ordered, integrated and consistent processing of them
- “Command Interpreter” is responsible for analyzing received commands to classify them into data access operations or group communication ones
- “Mass Data Management” is responsible for managing mass data and performing data access
- “Middleware” is responsible for transferring data between heterogeneous networks or systems

- “Group Management” is responsible for establishing and managing groups, for instance, to add or remove a member from a group
- “Synchronous Forward” is responsible for transmitting messages to group members in a synchronous way
- “Asynchronous Notification” is more complicated than Synchronous Forward. In this mode, each member is allowed to leave or return at any time. The “asynchronous notification” has to make sure that before he continues his work, he can receive all messages sent by the other members during his absence
- File system and database system is responsible for storing mass data in a distributed way.

### 3 Solutions to Key Technologies

#### 3.1 Management of Mass Data

The resources covered by our prototype system take on characters of large dataset size, geographic distribution and heterogeneity, which challenges existing data management methods. It is inevitable to seek for a new technology that should provide a convenient, efficient and safe solution [3]. Motivated by these considerations, we put forward a data grid-based mass data management method for Digital Virtual Human datasets. Fig. 3 shows the structure of the mass data management module.

Several main components of the Figure 3 are described below:

- Storage resource allocation

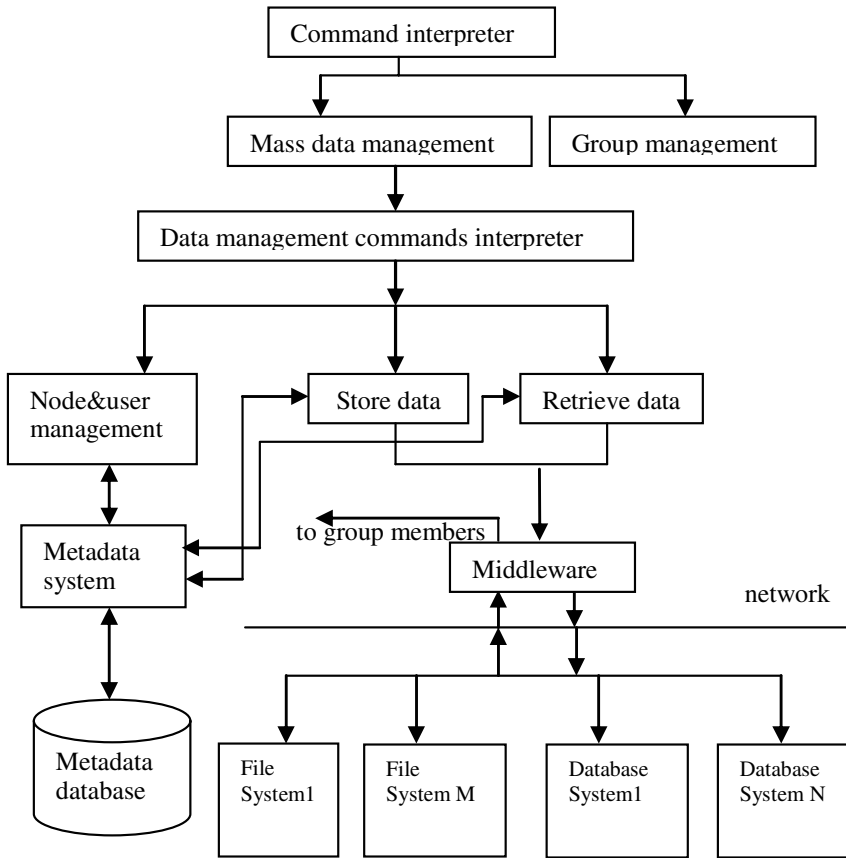
A straightforward allocation tactic is adopted here. Firstly all the available Digital Virtual Human datasets are divided into original datasets and derived datasets. The former is allocated to the servers in local area network, while the latter to PC nodes. By this way, all storage resources belonging to the grid are linked together to form a virtual storage environment, which enables a full utilization of resources.

- High speed data access

It can be achieved through three ways:

- a. Metadata:* Metadata aggregates information of grid users, nodes and data resources. It is designed to be application-specific which facilitates quick resources locating and provides a strong support for data query and data access.
- b. Data Compression:* Original datasets are compressed to reduce their size in order to shorten transfer latency and increase data transfer efficiency.
- c. Replica:* Replica mechanism means distributing several replicas of same datasets on multiple grid nodes, which saves data transfer time by accessing data according to “Nearest Access” policy [7].

- Data transfer between distributed, heterogeneous systems by virtue of middleware technology



**Fig. 3.** The structure of mass data management module

Middleware technology eliminates obstacles for information exchange between lower heterogeneous environments and provides application programs a relatively stable development environment. Consequently this technology is introduced here to solve our problem of transferring data between file systems and database systems or different networks, which would lessen our development work in a great deal.

The combination of above technologies enables us to realize a uniform management of distributed, heterogeneous mass data.

### 3.2 Implementation Methods of Application Collaboration Tools

The graphics tool of the prototype system stems from our previously developed nerve image processing software. To support cooperative work, the original version of the program is upgraded in several aspects that are discussed below:

- Capture group members' input commands through keyboard or mouse;
- Trace and find how these commands response in the program and locate the position where they are stored;
- Through synchronous forwarding or asynchronous notification, transmit these input commands to the same position of each group member's node, and make them perform the same operation as the sender does to make sure all members can view the same results.
- Allocate different colors to different group members, so each member's operating results can be displayed with his assigned color, which would increase collaboration perception among collaborators.
- To provide stronger support for collaboration work, other collaboration tools, such as speech tools, can be integrated into the system.
- "Member List" lists all group members. According to different demands, subgroups can be built timely and dynamically to perform discussion and communication on some specific topics.

## 4 A Case Study

In this section a case study is presented to illustrate how the prototype system works. Image segmentation and registration have always been an important but difficult job in the Digital Virtual Human modeling process. Take brachial plexus as an example, due to its complicated network topological structure (Fig. 4.a), pure automatic operations can impossibly generate absolutely correct results [8]. Both large amounts of tiny nerve bundles on nerve slices and complex nerve structures toughen the manual segmentation and registration operations and make it an ardent job. So we adopt a collaborative way to ensure validity and efficiency of the work.

Next, coupled with system functions, a whole "nerve image segmentation and registration" teaching process is described step by step.

### 4.1 Fetch Data

Firstly, the teacher starts the "data management" tool to download needed nerve slices from the grid to his local disk (Fig. 5). Then, the data are forwarded to all group members to make sure that every student would own the same copy. Next the nerve segmentation and registration is divided into two steps:

Step 1: Extract nerve contours on each slice by use of watershed-based algorithm, which would generate a series of binarized images (Fig. 4.b).

Step 2: Perform image registration on segmented images. That is to correspond nerve contours on adjacent slices to be ready for later three-dimensional reconstruction (Fig. 6).

During this process, each operation of the teachers and his corresponding speech explanation are bound together to be forwarded to all students. Then the students' nodes execute received commands automatically and result in the same output, which enables synchronous teaching and learning like in traditional classrooms.

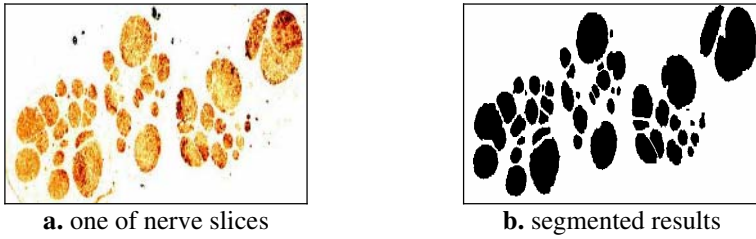


Fig. 4. One of nerve image slices



Fig. 5. The interface of the mass data management system



Fig. 6. One of the nerve correspondence results

## 4.2 Check and Tutor

Automatic operations cannot guarantee a 100-percentage correctness and wrong contour matches have to be corrected by human intervention. Under this circumstance, the teacher may appoint students to perform contour correspondence. That is to cancel those incorrect matches and give contours right connections manually. Taking characters of tutoring and work efficiency into consideration, this tutor function is implemented in an asynchronous way. Each operation of the students is sent to the teacher's node and saved in a settled buffer. When the teacher wants to check a student, he fetches the student's operations from the buffer and executes them. The graphics tool will facilitate the teacher to view the results and judge whether they are correct. If errors exist, he can communicate with that student through text or speech tool, point out his mistakes and help him to correct them.

## 4.3 Discuss

The teacher may arrange students to do some discussions to deepen their understanding of knowledge. During this process, discussions either between teachers and students or among students are allowed. Under this condition, both groups and group members are



changed dynamically. Application collaboration tools support direct interoperations among group members. Each one's operations would be forwarded to his group members and be executed on their nodes. Besides graphics tools, speech or whiteboard tools are also very important in supporting communications among group members.

In summary, during this whole teaching and learning process, both geographic distribution of teachers and students and heterogeneity of different platforms would possibly lead to data format mismatch or inconsistency. Consequently, middleware plays a critical role in shielding diversities of lower heterogeneous structures. According to system characteristics and requirements, it transforms data into corresponding format automatically, which enables a smooth information exchange between teachers and students.

## 5 Conclusions

In this paper, a Digital Virtual Human distance education prototype system is presented, which combines Digital Virtual Human model and distance education together and provides rich teaching means. Aiming to two key problems of the system-mass data management and application collaboration, corresponding solutions are proposed and developed. In order to store and manage Digital Virtual Human mass datasets effectively, data grid technology is introduced. It integrates distributed, heterogeneous storage resources into a virtual storage environment and provides efficient, safe and transparent data management methods, which facilitates data access in a great deal. Through upgrading existing nerve image processing software, the application collaboration among group members is realized, which provides stronger support to the distance education process.

The prototype system in local area network proves the validity and feasibility of the Digital Virtual Human distance education system, which lays a good foundation for our future work. Next this prototype system will be improved and spread from local area network to a wide area network.

## Acknowledgements

The work presented in this paper was supported by National High-Tech Research and Development Plan (Grant No.2001AA231031, 2002AA231021), National Key Basic Research Plan (Grant No. 2004CB318000) and National Special R&D Plan for Olympic Games (Grant No.2001BA904B08).

## References

1. Zhong, S., Niu, H., Li, H., Luo, S., Qin, D., Lin, Z.: Development and Applications of Chinese Digitized Virtual Human. Proceedings of No.208 Xiangshan Science Conferences: Development and Applications of Chinese Digitized Virtual Human, Beijing, (2003) 5-11

2. Lin, Z., Li, H., Liu, L., Jia, F.: Data Grid and Mass Data Management. Proceedings of No.208 Xiangshan Science Conferences: Development and Applications of Chinese Digitized Virtual Human, Beijing, (2003) 33-37
3. Liu, L., Jia, F., Zhao, G., Li, H., Lin, Z.: The Challenges for High Performance Computers from Digitized Virtual Human. China Basic Science, 2003(3) 44-47.
4. Chervenak, A., Foster, I., Kesselman, C., Salisbury, C., Tuecke, S.: The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets. Journal of Network and Computer Applications, 23(3) (2000) 187-200.
5. Wang, Y., Xiao, N., Ren, H., Lu, X.: Research on Key Technology in Data Grid. Journal of Computer Research and Development, 39(8) (2002) 943-947.
6. Hoschek, W., Martinez, J., Samar, A., Stockinger, H., Stockinger, K.: Data Management in an International Data Grid Project. IEEE/ACM Int. Workshop on Grid Computing (Grid'2000) (2000)
7. Vazhkudai, S., Tuecke, S., Foster, I.: Replica Selection in the Globus Data Grid. International Workshop on Data Models and Databases on Clusters and the Grid (DataGrid 2001), IEEE Computer Society Press, (2001)
8. Liu, L., Wang, S., Jia, F., Chen, R., Xiang, S., Liu, G., Li, H., Chen, Z., Chen, T., Chen, Z.: Computer-Aided Three-Dimensional Reconstruction of Microscopic Structure of Brachial Plexus Based on Consecutive Series of Slices. In proceedings of 8th international conference on CAD/Graphics, Macau, China, (2002) 46-51

# Towards Incompletely Specified Process Support in SwinDeW – A Peer-to-Peer Based Workflow System

Jun Yan<sup>1</sup>, Yun Yang<sup>1</sup>, and Gitesh K. Raikundalia<sup>2,1</sup>

<sup>1</sup> Faculty of Information and Communication Technologies,  
Swinburne University of Technology,

PO Box 218, Hawthorn, Melbourne, Australia 3122  
{jyan, yyang}@it.swin.edu.au

<sup>2</sup> School of Computer Science and Mathematics,  
Victoria University,

P.O. Box 14428, Melbourne City, Australia 8001  
Gitesh.Raikundalia@vu.edu.au

**Abstract.** Due to increased complexity and flexibility of processes and lack of modelling information, workflow processes are not always defined completely before their execution. Support for incompletely specified processes which require on-the-fly articulation of processes has become a desirable feature of workflow management systems. Unfortunately, this aspect is rather weak in contemporary workflow research. This paper reports innovative research on incompletely specified process support carried out in the context of SwinDeW, a peer-to-peer based decentralised workflow system. In order to extend the SwinDeW architecture and system functions seamlessly for supporting incompletely specified processes, a hierarchical process modelling and execution approach is presented in this paper. This approach supports stepwise elaboration of incompletely-specified processes on-the-fly. Further elaboration of a process is innovatively modelled as essential steps towards the process goal, thus being scheduled to execute as ordinary tasks.

## 1 Introduction

Workflow Management Systems (WfMSs) provide automated support for business processes, through the use of software. Over the past decade, as a solution to business process management, workflow technology has attracted intense attention from both researchers and practitioners and experienced tremendous growth. The adoption of WfMSs has, consequently, brought direct and indirect benefits such as reduction in costs and flow times, increase of quality of service and productivity, and so on. Although workflow research and practice have reached a certain degree of maturity, some severe problems remain unsolved. More specifically, the inappropriate use of the client-server architecture which provides centralised workflow coordination, and the lack of ability to support incompletely specified processes (*incomplete processes* for short) have become two major obstacles to the wide deployment of workflow technology in the real world. As a consequence, two growing trends of workflow technology have been observed. One is to build workflow management systems in a genuinely distributed way to reflect workflow's inherently distributed feature more

naturally. The other is to develop enabling techniques and mechanisms for support of incomplete processes.

Many efforts have been devoted to addressing issues related to the first trend, from adding more distribution to client-server based workflow systems to adopting new computing technologies such as peer-to-peer (p2p) technology [2] for workflow support. In particular, the authors have conducted intensive research aiming at combining workflow and p2p technologies to offer decentralised workflow support. An innovative workflow approach, known as SwinDeW (*Swinburne Decentralised Workflow*) [12, 13], has been presented to support completely specified processes (*complete processes* for short) in a p2p environment. Regarding incomplete process support, although some preliminary work has addressed the problems initially from the process modelling perspective, system coordination support for incomplete processes has been unreasonably ignored. Moreover, to the best of the authors' knowledge, no research has been carried out in addressing these two aspects jointly.

The distinct research reported in this paper was carried out in the context of the SwinDeW project. The major focus of this paper is to address the issues of incomplete process support in a p2p-based decentralised workflow environment from a system coordination viewpoint. To enable this, the following specific requirements for SwinDeW are raised: (1) An incomplete process definition needs to be divided into task partitions and task partitions need to be distributed to and stored by relevant peers properly. An incomplete part of a process should be allowed to instantiate before execution. The workflow represented by an incomplete part of a process should be allowed to be allocated to peers; and (2) Run-time incomplete process support needs to be carried out in a decentralised environment so that on-the-fly process elaboration can be performed at the right time and the right place by the right participant. Accordingly, in this paper, an innovative hierarchical workflow modelling and execution approach is presented, which allows for incomplete process specification at build-time and on-the-fly process elaboration at run-time.

The rest of this paper is organised as follows. Section 2 introduces SwinDeW briefly, including its decentralised system design and corresponding mechanisms supporting complete processes. In Section 3, the approaches to support stepwise process modelling and execution in the SwinDeW decentralised environment are presented. Section 4 then uses a research project as a sample to illustrate the authors' key ideas. After that, major related work is introduced in Section 5. Finally, Section 6 concludes the paper with the authors' contributions and outlines the authors' future work.

## 2 Background

As the traditional client-server based centralised architecture has faced more and more challenges in supporting workflow systems properly, p2p-based workflow systems have been recently recognised as one of the most strategic future directions for workflow research [4]. Based on the rationale that p2p reflects decentralised workflow much more naturally than client-server, the authors have presented an innovative SwinDeW workflow approach. This approach aims at investigation of process support

technologies for decentralised workflow systems based on the p2p, rather than client-server, distributed system architecture.

As shown in Figure 1, the novel framework of SwinDeW implies the presence of neither a centralised data repository for information storage, nor a centralised workflow engine for coordination [12, 13]. The system is defined as four layers. The top layer is a set of workflow participant software (WfPS) which defines the application-oriented semantics to fulfil workflow functions. Core services of the workflow system are provided at the service layer. The data layer consists of data repositories (DRs) which store workflow-related information such as process definitions and instance information. The communication layer facilitates direct communication among workflow participants.

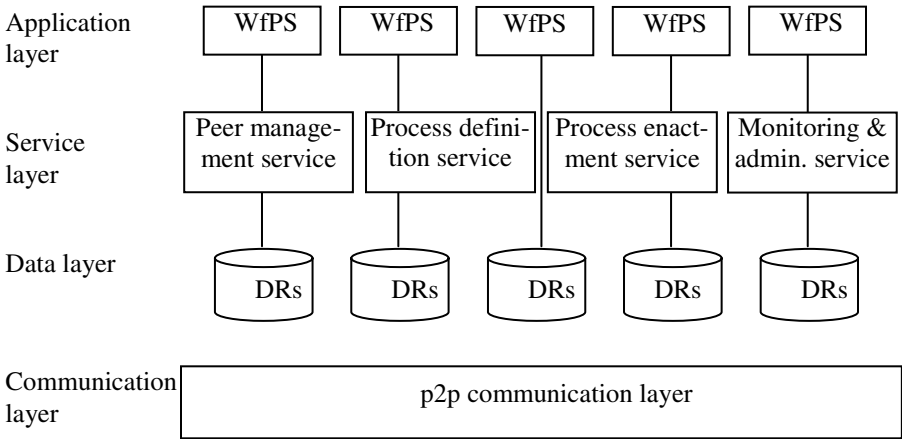


Fig. 1. Decentralised framework of SwinDeW

The basic working unit, known as a *peer*, consists of a WfPS and a set of data repositories. In most cases, each peer resides on a physical machine and is associated with a workflow participant. On behalf of the associated workflow participant, each peer is able to communicate with other peers directly to carry out functions. Given essential information and authority, a peer is a self-managing, autonomous entity whose ability to work both independently and collaboratively differentiates it from a client in the client-server architecture.

At build-time, the distinct data storage philosophy proposed in SwinDeW is named as “*know what you should know*” [12, 13], which is proposed in contrast to the conventional approaches in which each participant involved in a process knows either nothing or everything. Process definition data in SwinDeW are divided into a set of individual task partitions, each of which represents a single task in the context of the process. Thereafter, individual task partitions are distributed to relevant peers based on capability matching so that process definition data are stored in a distributed manner. By doing so, peers in the system have partial and essential knowledge, which enables them to collaborate in order to fulfil all the key functions of workflow execution.

Regarding run-time functions, SwinDeW mainly focuses on the issues of process instantiation and instance execution [12, 13]. The phase of process instantiation creates various task instances and determines performers of these task instances through peer coordination. All task instances are finally created at dispersed locations by peers actually performing them. Therefore, a process instance is represented by a network of peers performing various tasks in a certain order. Correspondingly, the execution of a process instance does not rely on a centralised engine to perform coordination. Peers communicate with one another to exchange control information and application data during process enactment. As each peer has knowledge about the task it is responsible for and its predecessor and successor peer(s), it can act independently to carry out different types of functions properly such as determining from whom they should receive notifications and data, when to start working, to whom they should pass work, and so on. In this way, the work is passed from one participant to another directly as predefined.

The above framework and mechanisms are designed to support complete processes. For demonstration and proof-of-concept purposes, a JXTA-based prototype has been implemented and the results so far are promising. The next logical step is to extend these mechanisms seamlessly for incomplete process support with unique features due to the p2p infrastructure. Some initial work of the authors is reported in [11, 12].

### 3 Hierarchical Process Modelling and Execution

To extend SwinDeW for supporting incomplete processes with on-the-fly task decomposition, a multi-tiered process modelling and execution approach is first presented in this section. Then, task decomposition is discussed. Finally, mechanisms for task execution are given.

#### 3.1 Hierarchical Modelling Approach

Modelling workflow means representing work activities in relation to each other. The classical approach in this view is best represented in “Hierarchical Task Analysis” (HTA) [7], where a workflow is described as a hierarchical structure of tasks and sub-tasks. In the real world, modelling of non-trivial processes is normally not a one-off occurrence and may experience several rounds. Initially, a process is defined as a network of tasks. Some of these tasks, known as *atomic tasks*, represent the smallest executable units of work. Normally, these tasks can be specified clearly and completely. Some other tasks, on the contrary, are simply modelled as black boxes with intakes and outtakes. Although each of these tasks, known as *composite tasks*, has a pre-determined contribution to the overall process, how a task is fulfilled remains uncertain for the time being. In most cases, a composite task represents a unit of work which is fulfilled by executing a set of sub-tasks. These sub-tasks, each of which can be either atomic or composite, are also partially ordered, forming a sub-process. Thus, a process is defined at multiple levels of abstraction. The process specification at a higher level of abstraction contains composite tasks which need to be decomposed in the process definition at a lower level of abstraction. Evidently, this approach adopts a

hierarchical modelling mechanism. Further process modelling work is to convert composite tasks into sub-processes. Hence, process modelling is carried out in a stepwise manner. Figure 2 depicts this top-down hierarchical modelling approach.

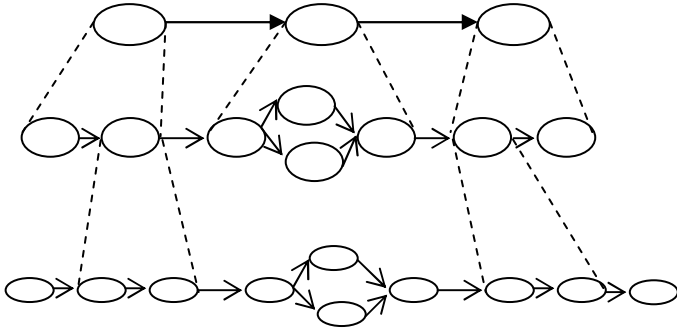


Fig. 2. The hierarchical modelling model

This approach rationalises the modelling work and reflects human’s behaviours naturally, especially when the process is complex and the modelling information is insufficient. In addition, this approach also provides multiple-level process reusability because multiple rounds of modelling generate multiple, reusable, model fragment drafts. A process model at a higher level of abstraction can be reused to generate various process templates at a lower level of abstraction. This feature allows the workflow system to deal with flexible workflow processes. Process instances with variations can be derived from the same process model at a certain level of abstraction and inherit some common features of the model.

### 3.2 Task Decomposition

To extend SwinDeW, a special managerial task, known as a *decomposition task*, is designed. Each decomposition task is associated with a composite task and in charge of the decomposition of this task. Figure 3 shows a decomposition task (*Decom*) and its position in the process where *Pre* and *Succ* indicate the preceding and succeeding tasks. The associated composite task,  $Compo(T_n)$ , in Figure 3 is finally decomposed into a sub-process which consists of sub-tasks  $t_{n,1}$ ,  $t_{n,2}$ ,  $t_{n,3}$  and  $t_{n,4}$ . The textual description of a decomposition task is as follows:

- *Description*: A decomposition task decomposes the associated composite task at run-time into a sub-process consisting of a set of partially ordered sub-tasks, each of which can be either atomic or composite.
- *Responsibility*: A decomposition task is carried out by an authorised person with special skills to model processes, such as a process engineer or a project manager.
- *Inputs*: The inputs of a decomposition task are very flexible. Normally, a decomposition task may take the outputs of the preceding tasks of the associated composite task, as its inputs. In addition, a decomposition task may have other inputs such as available resources, historical experience, specifics of the process, and so on.

- *Output*: The single output of a decomposition task (as an input of the composite task) is a detailed description of a sub-process, which is equivalent to the associated composite task in terms of the contribution to the overall process.
- *Incoming control*: A decomposition task receives notifications from the preceding tasks of the associated composite task upon their completion.
- *Outgoing control*: A decomposition task notifies the associated composite task upon its completion.

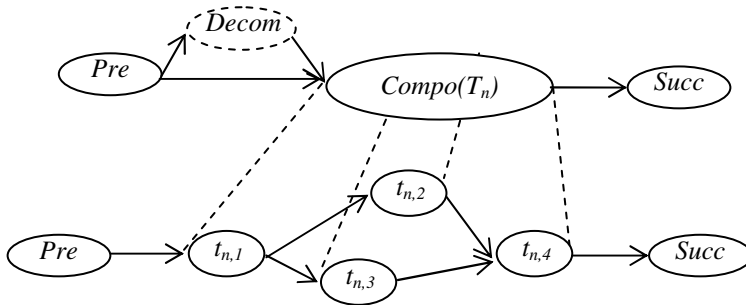


Fig. 3. A decomposition task and the associated composite task

Clearly, a decomposition task forms an *AND-JOIN* structure with the preceding tasks of the associated composite task. This is because the decomposition work should be completed before the execution of the composite task. Normally, a decomposition task should be carried out before the preceding tasks of the composite task are finished. This kind of decomposition is regarded as a “*pull*” model where a composite task is decomposed in advance. Hence, once the work is passed to the composite task, the sub-process resulting from the decomposition can be enacted immediately. However, in the worst case, the decomposition depends on the up-to-date status of process enactment and the peer associated with a decomposition task only starts when the preceding work has been done. This is the reason why a decomposition task receives the notifications from the preceding tasks of the composite task (see Figure 3). This kind of decomposition is regarded as a “*push*” model where the enactment of a decomposition task is triggered passively and may block the whole process.

With the support of the decomposition task, the definition of the composite task is temporarily stored together with the definition of the associated decomposition task, i.e., at those peers that are associated with authorised participants. As a result, the instantiation can be done with the same mechanism for atomic tasks. On behalf of an authorised person, a peer will create an instance of a composite task when required.

### 3.3 Task Execution

To enact a decomposition task in an organised manner, the associated peer may monitor the progress of process enactment through communication with other relevant peers and take various inputs into account. Additionally, analysis and modelling tools might be used to facilitate the decomposition performer. After the completion of a decomposition task, the associated peer notifies its single succeeding peer, i.e., the



peer associated with the composite task, and transmits an updated task description to advise the specifics of the composite task. The composite task is eventually enacted according to the updated task specification.

Once a composite task is decomposed into a set of sub-tasks, each sub-task is executed by a capable peer. Thus, the fulfilment of a composite task, i.e., the execution of sub-tasks, is achieved through coordination among relevant peers, including the peer creating the instance of this composite task and the peer(s) carrying out the sub-tasks. As a result, based on Figure 3, the peer network of the present instance is converted, as shown in Figure 4 where  $P_{pre}$  is a set of peers who execute the preceding tasks of  $T_n$ ;  $P_{succ}$  is a set of peers who execute the succeeding tasks of  $T_n$ ;  $P_{decom}$  is the peer who executes the decomposition task; and  $P_n$ ,  $P_{n,1}$ ,  $P_{n,2}$ ,  $P_{n,3}$  and  $P_{n,4}$  are peers who execute  $T_n$ ,  $t_{n,1}$ ,  $t_{n,2}$ ,  $t_{n,3}$  and  $t_{n,4}$ , respectively.

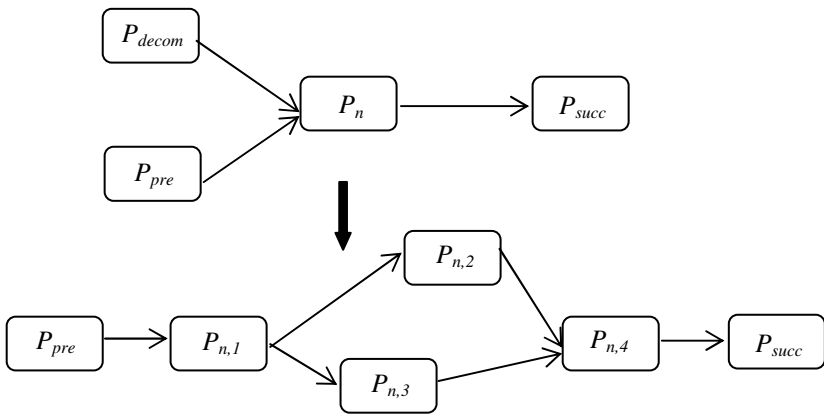


Fig. 4. Conversion of peer network for present instance

Decomposition can be performed at both the instance and process levels. Instance-level decomposition represents a provisional change and only takes effect in the present instance. Allowing instance-level decomposition reflects the fact that a flexible process may have multiple, variant instances. Each of the instances fulfils the composite tasks in a different way. On the contrary, process-level decomposition represents a permanent change to the workflow model and will be applied to all the instances created in the future. Permanent change is always associated with process evolution.

Instance-level decomposition is relatively simpler to handle, as the description of the sub-process is valid once only. In this case, peers  $P_n$ ,  $P_{n,1}$ ,  $P_{n,2}$ ,  $P_{n,3}$  and  $P_{n,4}$  simply use the local temporary client-server architecture to satisfy coordination requirements, where  $P_n$  acts as the server and  $P_{n,1}$ ,  $P_{n,2}$ ,  $P_{n,3}$  and  $P_{n,4}$  act as the clients. All the instances of the sub-tasks are created at peer  $P_n$ , and presented to  $P_{n,1}$ ,  $P_{n,2}$ ,  $P_{n,3}$  and  $P_{n,4}$  via a work list. Again, the selection of each of  $P_{n,1}$ ,  $P_{n,2}$ ,  $P_{n,3}$  and  $P_{n,4}$  is dynamic and negotiation-based. However, peers performing sub-tasks do not have direct interactions. Scheduling of the execution of the sub-tasks is done on the server side because

the logical structure of the sub-process only resides on  $P_n$ . This approach eases the implementation and management of sub-tasks because the client-server architecture reflects logically the hierarchical relationship among the peers. At the same time, this local centralisation does not have many side-effects because centralised coordination only occurs in a small scope temporarily.

Process-level decomposition is more complicated. To reuse the sub-process definition to create instances later, process data should be refreshed at the process level. The definition of sub-processes should be distributed properly and the existing process repositories of relevant peers may need to be updated. In addition, to execute the present instance properly, the network of peers should be reconstructed properly to reflect the changes. Using the composite task in Figure 4 as an example, execution of the decomposition results in a process-level conversion from  $T_n$  to  $t_{n,1}$ ,  $t_{n,2}$ ,  $t_{n,3}$  and  $t_{n,4}$ . The consequent operations of a process-level decomposition are described as follows:

- (1) The model of the sub-process, created by  $P_{decom}$ , is passed to  $P_n$ ;
- (2)  $P_n$  acts as the definition peer to partition and distribute the definition of the sub-process, with the mechanism discussed in [12, 13];
- (3)  $P_n$  instructs the relevant peers to create an instance of the sub-process, with the mechanism discussed in [12, 13]. The network of peers which consists of  $P_{n,1}$ ,  $P_{n,2}$ ,  $P_{n,3}$  and  $P_{n,4}$  is the result of this instantiation;
- (4)  $P_n$  advises  $P_{pre}$  to interact with  $P_{n,i}$  directly instead of contacting  $P_n$ ;
- (5) Similarly,  $P_n$  advises  $P_{succ}$  that  $P_{n,i}$ , instead of  $P_n$ , will interact with  $P_{succ}$  directly;

The above steps convert the peer network of the present instance. The following steps will update the process repositories of relevant peers:

- (6) Peers in  $P_{pre}$  update their process repositories of the new sub-process model. Namely, the post-conditions of the corresponding tasks are changed. Besides, peers in  $P_{pre}$  also propagate this update to other capable peers which have the relevant process definition, and let them know about this change;
- (7) Similarly, peers in  $P_{succ}$  update their process repositories of the new sub-process model. Namely, the pre-conditions of the corresponding tasks are changed. Besides, peers in  $P_{succ}$  also propagate this update to other capable peers which have the relevant process definition, and let them know about this change;
- (8)  $P_n$  deletes the definition of  $T_n$  and propagates this deletion within the corresponding virtual community because task  $T_n$  no longer exists for future instances;
- (9) Similarly,  $P_{decom}$  deletes the definition of the decomposition task and propagates this deletion within the corresponding virtual community because the decomposition task no longer exists for future instances.

After discussing the support for incomplete composite tasks, incomplete atomic tasks can be handled in a similar way. The assumption is that an incomplete atomic task can be treated as an incomplete composite task in SwinDeW. This is justifiable because an incomplete atomic task can always be grouped with other relevant tasks to form an incomplete, controlled, composite task naturally. These tasks are component tasks of the formed composite task. Another reason to use incomplete composite task support for incomplete atomic tasks is that incomplete atomic tasks are normally interrelated and need to be addressed as a whole.

In summary, meeting the requirements of supporting incomplete processes for SwinDew, both instance- and process-level decomposition can be carried out adequately, with the SwinDeW extension smoothly prototyped for proof of concept. The right person (authorised performer of the decomposition task) performs the decomposition work at the right time (either push or pull model) and at the right place (the peer associated with the performer), without bringing the whole system down. Due to space limits, other details such as inconsistency handling for currently running process instances and prototyping are not addressed here, and are available in [12].

## 4 Case Study

To illustrate the key ideas proposed in this paper, a CICEC (Centre for Internet Computing and E-Commerce) research project is used as an example. The experience obtained from this case shows that the approach proposed in this paper can support various scenarios of similar nature well. The project attempts to provide a leading-edge forum for the establishment, development, coordination, visualisation, testing and evaluation of Internet-enabled e-business ventures that will lead to successful, new e-business models and supporting techniques. Some related initiatives this research targets are development of (1) a suitable e-business modelling environment utilising the current and substantive knowledge and data of many e-business descriptions and models; and (2) a suitable wide-area workflow framework as infrastructure support for e-business processes.

Coincidentally, this project is conducted in the same way as a workflow process is executed in SwinDeW, although the management of this research project is not formally supported by a workflow management system. First, the project aims are achieved through the accomplishment of individual research tasks which have inherent relationships and should be performed in a certain order. Thus, the overall research project is easily modelled as a process. Second, this project involves several research teams which focus on various research tasks where communication and coordination among these teams are normally carried out directly between them.

Initially, only the major tasks of this project are specified. Tasks for e-business modelling research and workflow research are carried out by two research teams in parallel first, and then the task for integration needs to be carried out by another research team, to integrate the achievements. Obviously, at the early stage of the project, these three research tasks can only be modelled as composite tasks. Although the goals and expected outcomes of each task are expressed, how to specifically achieve the goals through some steps remains uncertain. The particular research schedule for each task can be gained only after some initial work such as a literature review has been done. In other words, the decomposition of composite tasks into sub-processes should be performed on-the-fly. For this reason, decomposition work is modelled explicitly as essential steps before the execution of the research tasks. The descriptions of three research tasks are firstly distributed to the leaders of three research teams, respectively, together with three extra decomposition tasks. Using the task of workflow research as an example, after some initial work (treated as the preceding tasks of the workflow research task) has been done, the leader of the workflow research team is able to specify how the research should be conducted. The correspond-

ing decomposition task is executed, resulting in a sub-process that consists of four sub-tasks which should be executed in sequence, namely system design, build-time functions, run-time functions, and prototyping. Then, these four tasks are assigned to the investigators in the workflow research team for execution. There is no evident difference between instance-level decomposition and process-level decomposition in this example, because this research project is a case-based process.

Practical experience has proved that stepwise process modelling and on-the-fly task decomposition naturally reflect the requirements of applications in the real world. In addition, the successful management of the research project also indicates that the approach proposed in this paper works well in supporting incomplete processes with uncertain composite tasks.

## 5 Related Work

Incomplete process support for workflow is an important area [1, 6]. Some work, although very limited, has been done in this area. WASA workflow [10], which aims at supporting flexible and distributed scientific workflows, presents a hierarchical workflow execution approach based on a set of states and accompanying state transitions for workflow instances. A complex activity may have a nested structure and activity models that are created using a set of activity modelling operations. Some other research (e.g., [3, 5]) focuses on human-centred solutions and argue that interactive enactment should be pursued more vigorously as a framework for flexible workflow modelling, allowing incomplete workflow models to emerge. Kumar and Zhao's research [8] represents a new approach to process design. It emphasises the dynamic perspective and is based on enumerating various tasks to be performed. The approach maximises the alternatives routes and enables dynamic routing during process execution to enhance flexibility. Mangan and Sadiq [9] develop a framework for specifying the process model from a standard set of modelling constructs and given process constraints. The constraints specification allows a process schema to be tailored to individual instance requirements.

These approaches address the issues of incomplete process support from the modelling technique rather than the system coordination support point of view. Although various mechanisms for process modelling are presented, aspects of system support for on-the-fly process articulation are rarely addressed. Moreover, these approaches are all based on the conventional client-server architecture. Relevant research on decentralised workflow environments is hardly ever conducted. This is where the work reported in this paper is founded.

## 6 Conclusions and Future Work

Incompletely specified process support has evidently become important in today's workflow research. In particular, decomposition of composite tasks on-the-fly is a typical case for incompletely specified processes because of increased process complexity and lack of information. In this paper, this difficulty is addressed from a system coordination support viewpoint, in the context of SwinDeW, which is a peer-to-peer based decentralised workflow system. A hierarchical process modelling and execution

approach is presented, which models and executes a process incrementally in a stepwise manner. The major distinction of this approach is modelling decomposition work as essential steps towards process goals. In this way, the build-time work and run-time work are seamlessly integrated, and the mechanisms used to support completely specified processes can be easily extended to support on-the-fly task decomposition.

In the future, further research on incompletely specified process support will be conducted. Issues of consistency and validity at both the instance and process levels will be further addressed. Other incompletely specified process aspects, such as task elaboration, dynamic process navigation, will also be further investigated.

## Acknowledgement

The research work reported in this paper is partly supported by Swinburne VC's Strategic Research Initiative Grant 2002-2004. It is also partly supported by the National Natural Science Foundation of China under grants No.60273026 and No.60273043.

## References

1. Aalst, W.M.P., Jablonski, S.: Dealing with workflow change: identification of issues and solutions. *Int. Journal of Computer Systems Science & Engineering* 15 (2000) 267-276
2. Aberer, K., Hauswirth M.: Peer-to-peer information systems: concepts and models, state-of-the-art, and suture systems. *Proc. of the 8th European Software Engineering Conf. (ESEC) and 9th ACM SIGSOFT Symp. on the Foundations of Software Engineering (FSE-9)*, 326-327, Vienna, Austria, Sept. 2001
3. Faustmann, G.: Configuration for adaptation - a human-centred approach to flexible workflow enactment. *Computer Supported Cooperative Work* 9 (2000) 413-434
4. Fischer, L. (ed.): *Workflow Handbook 2002*, Lighthouse Point: Future Strategies, (2002)
5. Håvard, J. D.: Interaction as a framework for flexible workflow modelling. *Proc. of the 2001 Int. ACM SIGGROUP Conf. on Supporting Group Work* (2001) 32-41, Boulder, USA, Sept./Oct. 2001
6. Jablonski, S., Stein, K., Teschke, M.: Experiences in workflow management for scientific computing. *Proc. of the 8th Int. Workshop on Database and Expert Systems Applications (DEXA'97)*, Toulouse, France, Sept. 1997
7. Kirwan, B., Aisworth, L.K.: *A guide to task analysis*. Taylor and Francis, London, 1992
8. Kumar, A., Zhao, L.: Dynamic routing and operational controls in a workflow management system. *Management Science* 45 (1999) 253-272
9. Mangan, P., Sadiq, S.: On building workflow models for flexible processes. *Proc. of the 13th Australasian Database Conf.* (2002)
10. Weske, M.: State-based modelling of flexible workflow executions in distributed environments. *Journal of Integrated Design and Process Science* 3 (1999) 49-62
11. Yan, J., Yang, Y., Raikundalia, G.K.: Critical issues in extending p2p-based SwinDew system for incomplete process support. *Proc. of the 8th Int. Conf. on CSCWD* (2004) 312-317, Xiamen China, May 2004
12. Yan, J.: *A framework and coordination technologies for peer-to-peer based decentralised workflow systems*. PhD Thesis, Swinburne University of Technology, Australia, (2004)
13. Yan, J., Yang, Y., Raikundalia, G.K.: SwinDeW - a peer-to-peer based decentralised workflow management system. Accepted by *IEEE Transactions on Systems, Man, and Cybernetics, Part A* (2005)

# A Flexible Workflow Model Supporting Dynamic Selection

Shijun Liu, Xiangxu Meng, Bin Gong, and Hui Xiang

School of Computer Science and Technology, Shandong University,  
Jinan, 250100, P.R. China  
{lsj, mxx, gb, hxiang}@sdu.edu.cn

**Abstract.** To improve the flexibility of workflow systems, efforts need to be made in the process of workflow modeling and system design. One of the enhancements would be the support of dynamic selection. In this paper, a workflow model based on the extended time interval Petri net with fired mark is presented, which satisfies the requirement of the dynamic treatment of workflows at the execution phase. A three-phase strategy and special node types are used to support dynamic selection, and the discussion about how they could improve the flexibility of the workflow systems is also given. Finally, a prototype system is presented to illustrate the effectiveness of the proposed model.

## 1 Introduction

Workflow is a process defined, operated and supervised by computer software. Such software is called workflow management system (WFMS) [1]. Mainly used at office automation system (OA) in the early years, more and more workflow systems are used in enterprises to support much more complex business processes such as those in integrated manufacturing systems recently. In fact, because of the complexity of the business processes in modern enterprises, WFMS has to be used.

Current workflow management systems have problems in dealing with both ad-hoc changes and evolutionary changes. While the trend is towards an increasingly dynamic situation where both ad-hoc and evolutionary changes are needed to improve customer services and to reduce costs [2]. Workflow management systems often need to define flow templates in advance, then the system would execute according to the predefined templates. Such system may be called rigid workflow. But the actual processes are often complex and changeful in practice, which demand the workflow to be flexible. That is to say, the flow would be changed in time according to the condition without interruption [3]. This is the flexibility of workflow.

Flexibility of a WFMS has two fundamental aspects: (1) The specification of a flexible execution behavior to express an accurate and less restrictive behavior in advance: flexible and adaptable control and data flow mechanisms have to be taken into account in order to support ad hoc routing and cooperative work at the workflow level. (2) The evolution of workflow models in order to flexibly modify workflow specifications on the schema and instance level due to process reengineering activities and dynamically changing situations of a real process [4].

So, flexibility is involved in both the modeling stage and execution stage of the workflow. To improve the flexibility of WFMS, dynamic selection of routing and instances is necessary. A workflow model based on time interval Petri net is presented in the paper, which satisfies the modeling requirement for the changing processes. Some strategies used in improving the flexibility of workflows are presented. Finally, we demonstrate the feasibility and effectiveness of the proposed model through a prototype workflow system implementation.

## 2 Related Work

There is a rich research literature on workflow, among which workflow modeling is the foundation of both theoretical studies and practical applications. The modeling methods used currently are mainly based on Activity network, Petri net, Theory of speech acts, Activity-state graph and Extended transaction model [5]. The existing workflow models fall short of description capability for enterprise applications, because of the insufficient capability of the semantics of workflow model to describe the complex process and the lack of rich enough definition of activity properties [6].

As a powerful modeling tool, Petri net can describe dynamic process of disperse events preferably and describe the sequential, concurrent and collision relations among events exactly. Petri net models the system graphically, which makes the system model intuitive and easily understood [7]. One of the many modeling approaches using Petri net [1] is WF-Nets [8], in which a Petri net can be used to specify the routing of cases. Activities are modeled by transitions and causal dependencies are modeled by places.

Many researches aimed at improving the flexibility of workflow systems. Flexibility can be categorized into two kinds:

- 1) Static flexibility. Which gives some freedom to users, allowing the users to choose the execution routing freely from several alternative routings before the workflow begins. Such flexibility can only deal with the routings defined in advance.
- 2) Dynamic flexibility. In some applications, adding an execution branch that could not be forecasted heretofore is necessary, which allow users to modify the workflow model to satisfy the process requirement during the execution [3].

Fan and Wu [5] pointed out that the essential reason of the shortage of the flexibility in workflow is that description mechanism does not reflect the actual condition of the application. They presented a workflow model composed of 3 kinds of connection arcs and 15 kinds of nodes too, which is based on harmony theory [5]. This model has powerful description capability, but too many node types may cause complicated situations during the system execution. Dynamic flexibility described in reference [3] is mainly about the changeable of topological structure of nodes. In fact, this flexibility only offers the real time modification of sub processes, which need a lot of effects of activity executors.

About routing flexibility, two approaches have been identified, namely flexibility by selection and flexibility by adaptation. Flexibility by selection is achieved by ensuring that there are a number of execution paths through the workflow process, such

that key decision making points are well represented. Flexibility by adaptation permits dynamic changes to workflows to include one or more new execution paths. [9]

In dynamic branching predictions, some approaches such as applying fuzzy logic are presented to select suitable branch according to actual condition and fuzzy rules [10]. Flexibility by dynamic adaptation is provided by the change and evolution of workflow models in order to modify workflow specifications on the schema and instance level due to dynamically changing situations of a real process [11].

Our approach based on dynamic selection addresses not only the dynamic selecting of routes, but also dynamic selection of activity instances based on matching. The dynamic selection will be done automatically or manually according to actual requirements. To gain the flexibility, a new Petri net-based model is presented and related strategies are discussed.

### 3 Workflow Modeling Based on Petri Net

#### 3.1 Transition-Time Interval Petri Net with Firing Mark

First quote a definition of Petri net [7]:

**Definition 1:** A Petri net is a triple  $(P, T, F)$ , where

$P$  is a finite set of places;

$T$  is a finite set of transitions;

$F$  is a set of arcs known as a flow relation. It is subject to the constraint that no arc connects two places or two transitions.

In order to increase the expressive power of a Petri net, some extensions are applied to the above-mentioned model. Timing can be added to a Petri net by naming all the places and transitions, and drawing up a table with minimum and maximum times for each transition to occur based on the time of arrival of the tokens at its input places. Timed Petri net is categorized into Stochastic Timed Petri net (STPN) and Deterministic Timed Petri net (DTPN). In STPN, the firing times are considered as random variables. If the transition times are deterministic, the Petri Net is called a DTPN. According to different timed characteristics, DTPN can be categorized into 3 kinds as follows [12]:

1) Transition – time interval Petri net: Every transition is associated with a time interval. The maximum (minimum) delay is defined as the below (up) bound of interval from the transition being authorized to occur. When the maximum delay times out, and is enabled from the minimum delay, the transition must occur.

2) Transition – duration Petri net: Every transition is associated with a time duration number. Transition occurs as soon as it is enabled, and moves tokens away from the preceding place. When the duration associated with transition is elapsed, tokens disappear, and new tokens generate in the successive place.

3) Place – duration Petri net: Every place is associated with a time duration number. Only when the delay associated with a place is elapsed, the tokens generated by transition occurrence are ready. The place can take part in enabling a transition then. And transition occurs as soon as being enabled.



Among them, transition–time interval Petri net is suitable for describing workflow, where transition denotes an activity which has time limit. In this paper, an extension of Transition–time interval Petri net is presented which adds a firing mark to every transition as the foundation of workflow modeling.

**Definition2:** Transition–time interval Petri net with firing mark (MTIPN)

MTIPN can be expressed as:  $MTIPN=(PN, TS, TE, M)$ , where:

1) PN is a general Petri net,  $PN = (P, T, F, u)$ ; P, place; T, transition; F, connection relation; u, tokens distribution in places.

2) TS, TE: Every transition ‘T’ can be assigned with a maximum and a minimum delay time.

3) M: Firing mark, is a function of  $T \rightarrow C$ , where set  $C = \{0,1\}$ . M denotes whether the fired transition is fired normally or compulsively for the maximum delay time out. M is attached to tokens.

Explanation: The normal firing condition is the same as that in general Petri net. But if transition is ready but not be fired for some reason, it will be fired compulsively when the maximum delay time is out, which insures the successive process carrying out on schedule. The two different firing conditions can be distinguished by firing mark. The transition fired compulsively but not executed really can be executed again by retrospect if necessary.

### 3.2 Workflow Model Based on MTIPN

**Definition3:** Workflow model based on MTIPN (WN)

WN is a tuple  $WN=(P, T, F, u, TS, TE, M, FD, A-VSet)$  where:

1) P is a finite set of places, and denotes conditions or states;

2) T is a finite set of transitions, and denotes activities;

3) F is a set of arcs, and denotes flows in workflow;

4)  $FD=\{\text{functional description of activity } t \mid t \text{ belongs to } T\}$ , and denotes functional description of activities;

5)  $A-Vset=\{2\text{-tuple (attribute-name, demand-value) set of activity } t \mid t \text{ belongs to } T\}$ , denotes specification of instance requirement;

6) u, TS, TE, M, have the same meaning as MTIPN.

Every activity in workflow is modeled as a transition in WN. Its execution condition is that all its input places are enabled, i.e., if all its input places contain at least one token. For transition that must be executed, its maximum delay is infinite, which means it could not be fired compulsively and must be executed. When transition is fired, tokens will be generated in its output places; this makes the successive transitions be executed for sure. In WFMS’s model, many building blocks such as AND-split, AND-join, OR-split and OR-join to specify workflow procedures [13], are expressed as transitions in WN:

- AND-split, transition transfers token in input place to every successive branch place, i.e., paralleling process.
- OR-split, transition transfers token in input place to one successive branch place, i.e., selection process.

- AND-join, transition will wait all tokens in each input place to reach, and will execute if this condition is satisfied, i.e., synchrony.
- OR-join, transition will wait tokens in one input place to reach, and will execute if this condition is satisfied, i.e., asynchrony.

To improve flexibility in describing practical applications, two characteristics are appended to WN. First, allow AND-split transition to transfer token to parts of successive branch places, which can be implemented by deleting some successive arcs dynamically. Second, judge the firing type of arrived tokens at AND-join transition, decide whether to execute current activity or retrospect by evaluating preceding activity. Fig. 1 shows an example of engineering change (EC) flow model based on WN.

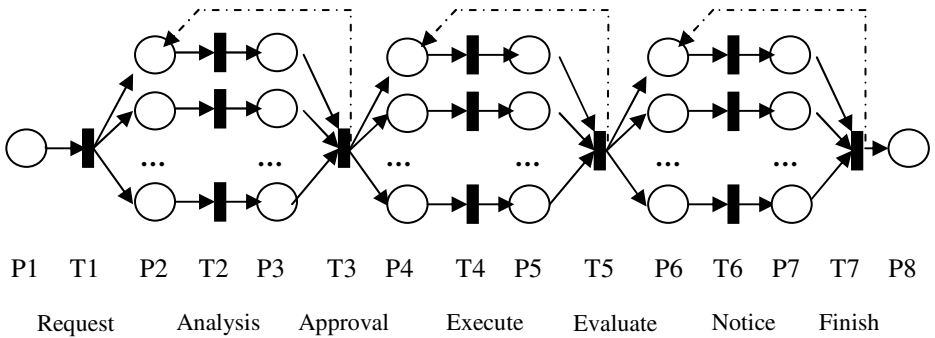


Fig. 1. EC workflow Petri net

Explanation: Places (states): P1: mistake finding, ready to submit EC request; P2: EC request is sent to persons related; P3: EC analysis finished; P4: EC order distributed to EC executor; P5: EC order executed; P6: EC notice sent to person related; P7: EC notice received; P8: EC flow finished.

Transitions (activities): T1: EC request submitting; T2: EC request analyzing; T3: EC analysis evaluating, locating data be changed, deciding EC level, making EC order; T4: EC order executing; T5: EC result evaluating and EC issuing; T6: accepting EC notice and feedback; T7: affirming EC flow integrity.

The whole EC flow is a sequential process, but has three parallel sub-processes: EC analyzing, EC executing and EC informing. Every transition has maximum and minimum delay. For transitions T1, T3, T5 and T7 must be executed, the maximum delay of them is infinite, while each transition in three sets of parallel transitions T2, T4 and T6 has a certain delay (activity executing time), and will be fired compulsively if the maximum delay times out. Some parallel branch not really finished can be judged by firing mark and be evaluated in transitions T3, T5 and T7. If such branch has no essential effect on the next activities, it might be ignored and current activity begins to be executed. If such transition branch must be accomplished, it can be enabled again (shown by broken line).

Based on WN, a workflow with time limit and dynamic control can be described, and be attached with functionality and specification description if necessary. But to gain active flexibility by dynamic selection, some special strategies are needed.

## 4 Strategies to Enhance Flexible of Workflow System

### 4.1 Three-Phase Modeling

Many workflow systems use templates in flow definition for its simple, intuitionistic, easy to use, and one flow template can be instantiated many times, which reduce the repeated work in flow definition. Template is composed of nodes and arcs, where nodes stand for activities and arcs stand for relations between activities. An actual flow is an instance of template, which is driven and controlled by workflow engine.

But this approach does not do well in changeful flow, where the flow lies on dynamic selection during execution. For example, what need to be changed in EC execution can only be known after EC analysis is finished.

To solve this problem, we introduce a three-phase modeling method, which is still a template-based method but has more flexibility by dividing the workflow modeling procession into functional modeling, specification defining and runtime phases (Fig 2):

#### 1) Functional model

In this model, only functional descriptions of workflow are given, including definition of the functionality of each activity, flows of process, and some selection nodes if needed. For the functionality of whole process has been defined, the workflow model could be evaluated by simulating and be stored as a template file.

#### 2) Specification model

When a workflow template is initialized, some decided activities would be assigned by actual parameters. At the same time, dynamical activities, which should be decided during runtime, will be attached with a specification of requirements and selection criteria.

#### 3) Runtime model

During the execution of a workflow, two kinds of dynamic selections will occur: choose a right flow branch and select a most appropriate activity instance. A right

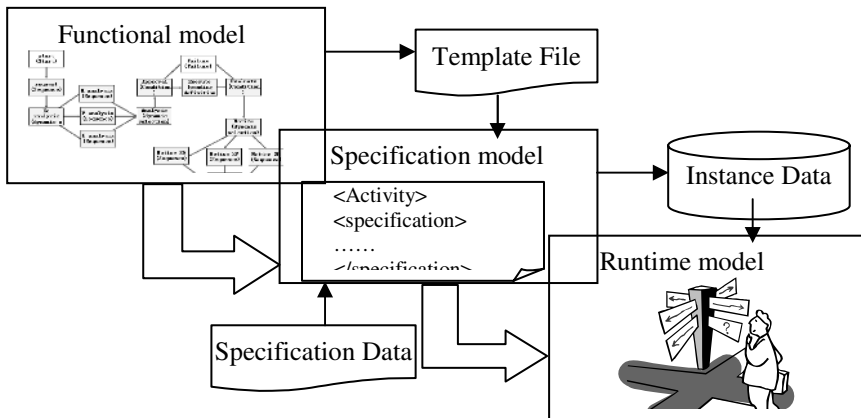


Fig. 2. Three-phase workflow modeling

flow branch is decided by the conditional rules pre-established and the results of preceding activities. An activity instance will be selected by matching based on functionality description and requirement specification.

## 4.2 Node Types

To enhance the flexibility of workflow, we bring forward some special node types and rules. At the same time, the node types should be less for the sake of reducing difficulty in system learning and using. There are seven types of nodes in our system: start, finish, failure, sequential, condition, dynamic selection, pending definition. By using these types together, the four basic routings defined by WPMC [13]: sequential, parallel, conditional and iteration routing can be described.

Start, finish and failure nodes are three default nodes of every template, denoting the beginning, ending and aborting of a workflow respectively.

Sequential nodes denote generic activity, which is executed in sequence and without conditions. Every activity has an overtime startup switch. If the switch is on, the workflow will be driven forward by engine when the activity is overtime, regardless whether the activity is accomplished or not, and a special tag will attach to the activity on this condition. This mechanism ensures successive activities without waiting too long for some abnormal conditions. This switch turns on for parallel nodes and turns off for sequential nodes in default.

Condition nodes are used in dealing with routing selections, which have true and false way outs defined at template definition, supplying a kind of static flexibility. Condition nodes can describe iteration and retrospect routing too, by introducing flow to a preceding node.

## 4.3 Special Nodes and Logic for Dynamic Selection

Dynamic flexibility is mainly embodied by dynamic selection and pending definition nodes.

Dynamic selection nodes include dynamic branch nodes and dynamic activity instance selection nodes.

When a dynamic activities selection node is executed, the most appropriate instance is selected from several candidate instances, which has the same functionality according to the functionality description in functional model and specification defined in specification model.

Dynamic branch nodes are mainly used in dealing with the parallel process splitting and joining. A dynamic split node has many parallel branches when being defined, all of which need to be executed in default, but only part of them would be executed in flow execution, which is decided by the executor of the dynamic branch nodes. The difference of dynamic branch nodes and conditional nodes is that the former can be chosen dynamically during flow execution while the latter is fixed in template definition.

A dynamic join node has many parallel preceding branches when being defined. When the current node is ready, some preceding branches may not be really accomplished but pushed forward compulsively for overtime. The executor of current node should evaluate such preceding branches and decide whether ignoring them or requiring them to resubmit. If no preceding branches need to be executed again, the current node can be executed, or else, the current node is suspended and preceding branches

resubmitted driven by engine again. If parallel branches joined in a dynamic branch node more than one, the overtime startup switches of them turn on.

Pending definition node is another type of special node. Many activities could not be forecasted and defined exactly at template definition phase. Moreover, many activities depend on the result of preceding activities execution. What should do in these activities is only decided by actual flow execution, which can be substituted by defining a pending definition node. The action of pending definition node is to complete sub-flow needed by executor who has privilege of flow definition. New sub-flow is defined and inserted into the current instance in the flow execution.

To obtain feedback of activity in real time, a backtracking mechanism is needed. Iteration routing can be treated as a kind of static feedback, where backtracking is achieved by arcs back to preceding activities. Using dynamic branch join node and overtime compulsively pushing mechanism, dynamic backtracking can be obtained. There may be no obvious connection arcs, but driven by engine in flow execution.

### 5 Implementation

We designed a prototype workflow system based on the proposed model, which is part of the product design management system for a machine tool manufacturer. The prototype system composed by the following three parts: engine, modeling tool and client. The prototype was developed using .Net, C# and Asp.net technology.

As shown in Figure 3, EC analysis includes some parallel processes: manufacturing analysis, technological analysis, stock analysis, etc., but not all of them need to be executed in an EC flow, so EC analysis activity is modeled by a dynamic branch node, which needs to be executed and decided by the executor of this node. Which EC order being executed depends on the result of EC analysis, which could not be known exactly at the time of the template being defined. This activity can be modeled by

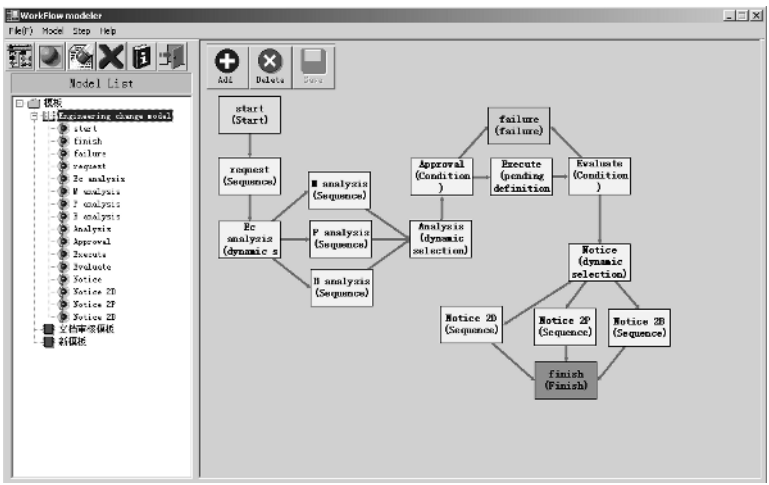


Fig. 3. An EC template modeling demonstration

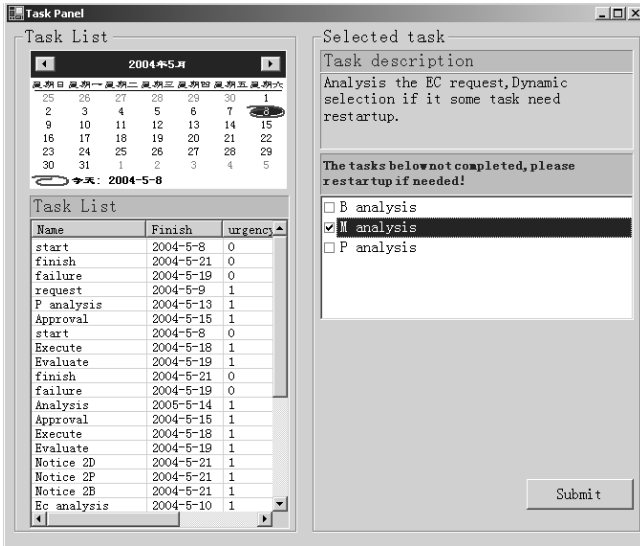


Fig. 4. Dynamical treating in client

pending definition node. When the activity is executed, the actual sub-flow is defined and inserted into the current instance. The overtime compulsively pushing mechanism of some parallel branches ensure the synchronization of these branches. Figure 4 shows a dynamic treating in client.

## 6 Conclusion

The flexible workflow model and design strategies presented in this paper can satisfy the demands of describing complex and changeful processes. Moreover, it obtained by less node types and simpler rules. The model's feasibility is demonstrated by Petri net theory and our prototype system. However, the system still requires further improvements to obtain better performance.

## Acknowledgement

The authors would like to acknowledge the support provided by the National High Technology Research and Development Program of China (No. 2003AA414310).

## References

1. Luo, H., Fan, Y., and Wu, C.: The Overview of Workflow Technology. Journal of Software. 11(7) (2000) 899-907 (in Chinese)
2. Van der Aalst, W.M.P.: How to Handle Dynamic Change and Capture Management Information: An Approach Based on Generic Workflow Models. Computer Systems Science and Engineering. 15(5) (2001) 267-276

3. Min G.T., Feng, T.: Dynamic Flexible Workflow Management System Based on Middleware Technology. *Computer Integrated Manufacturing System*. 8(8) (2002) 655-660 (in Chinese)
4. Joeris, G., Herzog, O.: Towards Flexible and High-Level Modeling and Enacting of Processes. *Proc. of the 11th Int. Conf. on Advanced Information Systems Engineering (CAiSE'99)*. Heidelberg, Germany. (1999)
5. Fan, Y., Wu, C.: Research of a Workflow Modeling Method to Improve System Flexibility. *Journal of software* 13(4) (2002) 833-839 (in Chinese)
6. Fan, Y., Wu, C.: Current State and Development Trends of Workflow Management Research and Products. *Computer Integrated Manufacturing System*. 6(1) (2000) 1-7 (in Chinese)
7. Song, Y., Chu, X., and Cai, F.: Research of The Process Modeling of Real-time Concurrent Design based on Time Petri Net. *Computer Integrated Manufacturing System*. 5(6) (1999) (in Chinese)
8. Van der Aalst, W.M.P.: Structural characterizations of sound workflow nets. *Technical Reports*. 96/23. Eindhoven: Eindhoven University of Technology, (1996)
9. Halliday, J., Shrivastava, S., and Wheeler, S.: Flexible Workflow Management in the OPENflow system. In *Fifth IEEE International Enterprise Distributed Object Computing Conference*. (2001). <http://citeseer.ist.psu.edu/halliday01flexible.html>
10. Zirpins, C., Schütt, K., and Piccinelli, G.: Flexible Workflow Description with Fuzzy Conditions. <http://citeseer.ist.psu.edu/zirpins02flexible.html>
11. Joeris, G.: Defining Flexible Workflow Execution Behaviors. *Enterprise-wide and Cross-enterprise Workflow Management*. (1999) 49-55
12. Zhang X.: Actor Model:A Multimedia Data Presentation Model. *Journal of software*, 7(8) (1996) 471-480 (in Chinese)
13. Wfmc: Workflow Management Coalition Terminology & Glossary. [http://www.wfmc.org/standards/docs/TC-1011\\_term\\_glossary.pdf](http://www.wfmc.org/standards/docs/TC-1011_term_glossary.pdf)

# Temporal Logic Based Workflow Service Modeling and Its Application

Huadong Ma

School of Computer Science & Technology,  
Beijing University of Posts and Telecommunications, Beijing 100876, China  
mhd@bupt.edu.cn

**Abstract.** This paper proposes an approach to modeling workflow services based on temporal logic. This model supports the formal specifications of various entities and workflow services for a workflow system. The model can specify the workflow process and its evolution, which are difficult to be supported by the previous models. This model is used to design a workflow specification language. Based on this language, we have developed a workflow service editing tool, which is the kernel of an interactive workflow design environment.

## 1 Introduction

Workflow is a kind of business procedures to be completely or semi-automatically executed. There are two key parts in a workflow management system [1], i.e., workflow modeling and workflow engine. Workflow modeling provides a build-time environment to define, analyze and manage the workflow service. Generally three kinds of models are concerned: organization model, data model and process model [2]. Workflow engine is a run-time environment for creating, executing, and managing a workflow service. It is a complex work for developing a workflow management system, which should efficiently support the design and execution of a workflow service. Thus, it is necessary to formally specify workflow systems. Especially for the workflow modeling, a powerful step-wise refinement design environment must be available, but few previous efforts aimed at solving this problem. Although the previous models are good at specifying the activities and synchronization among them, they cannot specify the workflow process and its evolution dynamically.

Workflow system requires a powerful workflow design environment to be used easily. XYZ system provides an effective way to model the process of a workflow service dynamically. XYZ system is a powerful CASE environment, whose kernel is a temporal logic language XYZ/E [3, 4]. The philosophy of XYZ was used to design multimedia scripts [5, 6] and hypertexts [7], for the specification of various entities and the procedure of designing a script in multimedia and hypertext systems. CASE environment provided by XYZ system is also a good platform to develop a workflow editing system. This environment supports not only the specification of various objects in workflow system but also the specification of the workflow service, activities, concurrent processes, step-wise refinement design, and so on. This paper proposes a workflow specification model based on temporal logic, called TLWS model. Using this model, we can specify modeling activities, designing and scripting a workflow



service in a unified framework. An interactive workflow design system based on this model has been designed and developed.

The rest of this paper is organized as follows. Section 2 reviews the related work; Section 3 describes the proposed model; Section 4 presents a workflow specification language based on the proposed model; Section 5 discusses the implementation issues for interactive workflow system; Finally, Section 6 concludes the paper and discusses future work.

## 2 Related Work

### 2.1 Related Workflow Models

The main function of workflow model is the specification of temporal synchronization among activities. The activity can be a single atomic activity or composed activities. The presentations of activities are combined by the activity operators in a workflow service. There have been some efficient methods to specify the workflow service. In general, existing workflow models are as follows:

- *Graph model*: A workflow is defined as a directed acyclic graph with a beginning node and an ending node [8]. A node represents an activity, and an edge represents a transition.
- *Petri-Net model*: Place and transition in Petri-Net are used in specifying workflow service process, where a place represents a control node and a transition represents an activity. Ellis and Nutt specify workflow process by Information Control Net (ICN) [9]. Two kinds of control nodes (and, or) specify the control relations among activities. A specific Petri-Net model for workflow is Workflow Net (WF-Net) [10]. A WF-Net is with a source place and a sink place. A detail survey on Petri-Net based workflow model is provided in [11].
- *Event-Condition-Action (ECA) model*: An ECA rule is used to specify service process in [12]. The ECA rule is a service rule. That is, when an event occurs, if the condition is true, the action is executed, otherwise alternative action is executed.
- *CTR model*: Davulcu *et al.* used Concurrent Transaction (CTR) Logic to specify and analyze the workflow service process precisely [13].
- *Hypermedia model*: Haake and Wang used hypermedia structure to describe workflow service [14], in which a node is the specification of activity, and a link represents a relation between activities.
- *Speech Act model*: Medina-Mora *et al.* proposed a workflow model based on speech act theory [15], and it defines workflow process from client and server, respectively. The workflow service consists of workflow cycles.

### 2.2 Elementary Workflows

A workflow service can be composed of some elementary workflows. Generally, there are following eight elementary workflows (Fig. 1): Sequence, AND-join, AND-

split, Condition-branches, Executive-OR-join, Time-based, Parallelism & Time-based Delays, and Event Trigger.

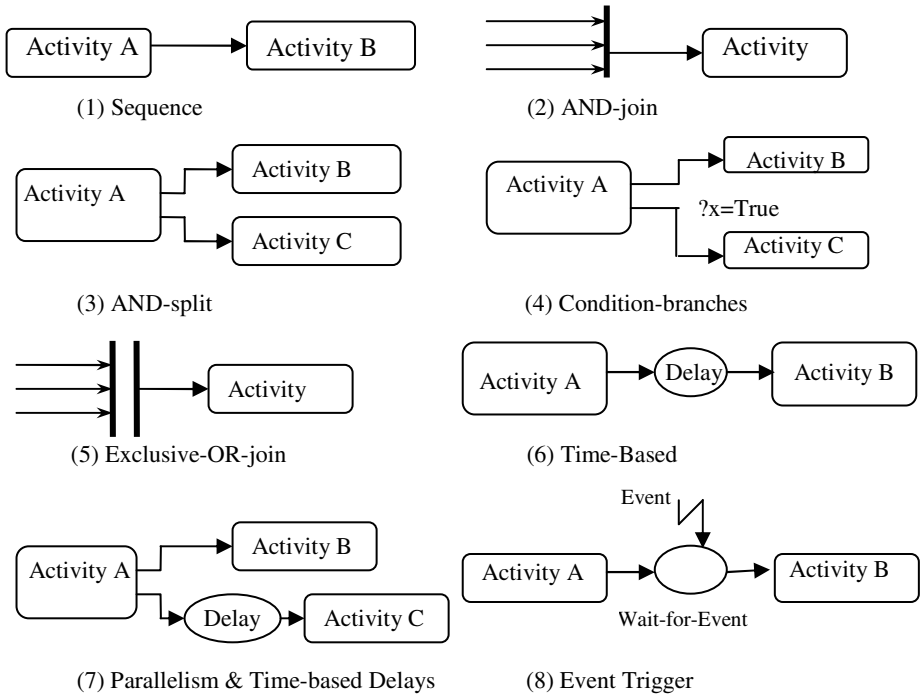


Fig. 1. Elementary workflows

2.3 Temporal Logic and XYZ System

Based on the linear temporal logic theory, Tang designed a temporal logic based CASE environment, XYZ system, which can support various ways of programming [3,4]. XYZ system is a family of programming languages extended by XYZ/E.

In XYZ/E, the statement is called condition element (c.e.). In order to describe non-determinative and concurrency, the form of c.e. is as follows.

$$LB=y_i^{\wedge}P \Rightarrow @ (P_{rj1} \& \dots \& P_{rjk})$$

$$LB=y_i^{\wedge}P \Rightarrow @ (P_{rj1}) \& \dots \& @ (P_{rjk})$$

where  $y_i$  is the current label,  $P$  is the condition for executing the temporal logic formula in the right part,  $@$  is one of the following temporal operators:  $\$O$  (Nexttime),  $\langle \rangle$  (Eventually),  $[]$  (Always),  $\$U$  (Until).

Two kinds of the above combinations are meaningful in the view of programming. The first is “selection statement”, and its form is as follows.

Selection statement:

$$LB=y_i^{\wedge} P \Rightarrow \$O[Con_1^{\wedge} ExeAct_1 \vee \dots \vee Con_k^{\wedge} ExeAct_k]$$

where  $Con_i$  and  $ExeAct_i$  represent the part of condition and the part of action, respectively. Assume that the forward labels of all  $ExeAct_i$  are EXIT.

The second is “parallel statement” which describes the concurrency in programming, and its form is as follows.

Parallel statement:

$$LB=y_i^{\wedge}P \Rightarrow \$O (P_{rj1}^{\wedge} \dots^{\wedge} P_{rjk})$$

$$WHERE \parallel [P_{rj1}, \dots, P_{rjk}]$$

where “ $\parallel [P_{rj1}, \dots, P_{rjk}]$ ” represents  $\parallel [P_{rj1} \$ \vee \dots \vee P_{rjk}]$ .

The message passed from one process to another is implemented by using channel operations. In XYZ/E, there are two channel operation commands: output command (write channel)  $Ch!y$  and input command (read channel)  $Ch?x$ , where  $Ch$  is the name of channel from the input process to the output process,  $y$  is an output expression of the output process, and  $x$  is an input variable of the input process.

### 3 Temporal Logic Based Workflow Service Model

In the workflow system, activities can be classified into two categories: the basic activities and the composed activities. Roles are various actors in the execution of workflow service, and they handle the attached document with a workflow service.

**Definition 3.1.** *Activity* is defined as follows:

- (1) atomic activity  $(r, t)$  is an activity in which the role  $r$  is handling the task  $t$ ;
- (2)  $\omega$  (null activity) is a special activity whose duration time is not 0, but there is no task to handle;
- (3) *delay* (with a parameter  $d$ ) is a special activity whose lasting time is  $d$ , but there is no task to handle;
- (4) *composed activity* constructed by an activity expression is also an activity.

**Definition 3.2.** *Activity operator.* If A, B and C are activities, the activity operators between activities are defined as follows:

- (1) *Sequence:*  $A;B$ .  $A$  is presented first, then  $B$ . The expression ends when  $B$  ends.
- (2) *AND:*  $A^{\wedge}B$ .  $A, B$  start simultaneously, and the expression ends when both of activities end.
- (3) *OR:*  $A \vee B$ .  $A$  or  $B$  starts, and the expression ends when either of activities ends.
- (4) *Case:*  $(case_1^{\wedge}A_1) \vee (case_2^{\wedge}A_2) \vee \dots \vee (case_n^{\wedge}A_n)$ . If  $case_i$  is True, the activity  $A_i$  is executed ( $1 \leq i \leq n$ ).
- (5) *Loop:*  $A^*m$ . Repeat  $m$  times to present  $A$ .

**Theorem 3.1.** The set of operators consisting of  $\{Sequence, AND, OR, Case, Loop\}$  is complete.

**Proof:** we can easily prove that all of eight elementary workflows in Fig.1 can be represented by the operators provided by Definition 3.2.

**Definition 3.3.** *Activity expression* is defined as follows:

- (1) An activity itself is an activity expression;

- (2) If  $X$  is an activity expression,  $(X)$  is an activity expression;
- (3) The result of operating activities (*Sequence, AND, OR, Case, Loop*) is an activity expression;
- (4) In an activity expressions, the orders of priority of operations are as follows:  $()$ ,  $*$ ,  $\wedge$ ,  $\vee$ ,  $\vee'$ ;
- (5) All activity expressions can be formed by using the above rule (1) - (4).

**Definition 3.4.** *Temporal Logic based Workflow Specification* (TLWS) model is defined as a six tuple,  $TLMS = (\Sigma, F, R, T, O, S_0)$ , where  $\Sigma$  is a set of scene states,  $R$  is a set of roles,  $T$  is a set of tasks,  $S_0$  is the start scene state.  $F$  is called as a set of scene state transition rules, and a rule is formed as:  $lb = S_i \wedge P_i \Rightarrow @_i(Q_i \wedge lb = S_j)$ ;  $O$  is a mapping from rule to activity expression:  $F \rightarrow \{activity\ expression\} \cup \{\omega\}$ .

Compared with the previous models [8-15], TLWS model has some excellent features. That is, TLWS model supports the specifications of following aspects: abstraction of activities for a workflow service; synchronizations among activities; process of workflow service; step-wise refinement design of a workflow service, thus the model supports the specification of workflow process and its evolution. Those features make TLWS a powerful model. We compare it with the previous models [2] in Table 1.

**Table 1.** Comparing workflow models

<b>Characteristic</b>	<b>Graph</b>	<b>Petri Net</b>	<b>ECA</b>	<b>CTR</b>	<b>TLWS</b>
Semantic	No	Yes	No	Yes	Yes
Activity abstraction	No	No	No	No	Yes
Activity synchronization	Yes	Yes	Yes	Yes	Yes
Process specification	Yes	Yes	Yes	Yes	Yes
Refinement design	No	No	No	No	Yes

## 4 Workflow Specification Language

Using the proposed TLWS model, we design a workflow specification language WSL and develop a workflow service editing tool. A WSL statement is a condition element. Tasks in WSL are defined as document forms. Activity synchronization can be specified by some transition rules. An activity is represented as  $(r, t)$  where the role  $r$  is handling the task  $t$ . Activity operations between  $(r1, t1)$  and  $(r2, t2)$  in WSL are defined as:

- (1) Sequence:  $(r1, t1); (r2, t2)$ .
- (2) OR:  $(r1, t1)|(r2, t2)$ .
- (3) AND:  $(r1, t1) \wedge (r2, t2)$ .
- (4) Case:  $(case1 \wedge (r1, t1)) \vee' (case2 \wedge (r2, t2))$ .
- (5) Loop:  $(r1, t1) * m$ .

The workflow service specified by WSL is similar to a XYZ/E program. A complex workflow can be designed in the way of step-wise refinement. The workflow can

be divided into several steps; and one step can be further divided into sub-steps; and the subdivisions go on until the workflow is composed of elementary workflows.

The general form of a WSL workflow service is as follows.

```
%WORKFLOW Wf_Name== [] [
  %VAR[Global Variables Description];
  %PROC [
    Proc1()==[
      //a procedure
      %LOC[Local Variables Description];
      %STM[Procedure Control Stream]
      .....]
  %LOC[Variables Description];
  %STM[Main Workflow Control Stream]
]
```

In order to illustrate the procedure of designing a workflow service by WSL, we define the workflow service consisting of some steps at first. Its typical specification is as follows.

```
%STM [lb=START=>$0lb=s1;
  lb=s1=><>(step1()^lb=s2);
  lb=s2=><>(step2()^lb=s3);
  .....
  lb=s5=><>(step5()^lb=s6);
  lb=s6=>$0lb=STOP]
```

Next, we continue to specify the steps. For example, *step2* is divided into two sub-steps, and the corresponding specification of workflow is as follows.

```
%STM [lb=START=>$0lb=s1;
  lb=s1=><>(step1()^lb=s2);
  lb=s2=><>$0lb=s21;
  lb=s21=><>(step21()^lb=s22);
  lb=s22=><>(step22()^lb=s3);
  lb=s3=><>(step3()^lb=s3);
  .....
  lb=s5=><>(step5()^lb=s6);
  lb=s6=>$0lb=STOP]
```

The design of workflow can be decomposed until every step and sub-step is divided into elementary workflows. Using WSL, we can define the roles, tasks and temporal relations between the activities in the workflow service.

For example, we can specify the device purchase process in an enterprise as follows.

```
Workflow WF_Device_Purchase()==[
  %LOC[Role applicant, amanager, president,
    pmanager, buyer;
  Task request, signature, permit, audit, order;
  Timer d1=5;
  Var cost;]
  %STM[ lb=START=>$0lb=s1;
  lb=s1=>$0{(applicant, request)}^$0lb=s2;
  lb=s2=>$0{(amanager, signature)}^$0lb=s3;
  lb=s3^(cost>=1000)=>$0{(president, permit)}^$0lb=s4;
  lb=s4=>$0{(pmanager, audit)}^$0lb=s5;
  lb=s5=>$0{(buyer, order)}^$0lb=s6;
  lb=s6=>$0lb=STOP]
]
```

In this example, the device purchase workflow *WF\_Device\_Purchase* includes five tasks: *request*, *signature*, *permit*, *audit*, *order*. In the specification of stage stream, five roles (*applicant*, *amanager*, *president*, *pmanager*, *buyer*) are defined to handle the tasks in different stages. First, *applicant* submits the purchase request, and his manager *amanager* checks the purchase request. If the cost is greater than 1000\$, the purchase request must be submitted to *president* for approval. Then, the request is transferred to the purchasing manager *pmanager* for verifying the budget. Finally, the purchase request is executed by the *buyer* and the workflow is terminated.

## 5 Interactive Workflow Design Environment

### 5.1 Architecture of Workflow Design System

The interactive workflow design system includes the following main parts (see Fig.2):

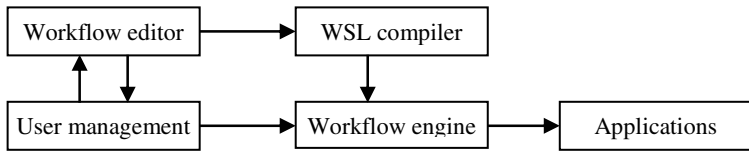


Fig. 2. Architecture of workflow design system

*Workflow editor* provides an interactive environment to define the workflow process in a step-wise refinement way. The result of designing a workflow service can be automatically converted into WSL file.

*User management* provides an interactive tool to edit the structure of users in a workflow service. The structure can be defined as a tree in which leaf node is specific user role and non-leaf node represents a department consisting of a group of users.

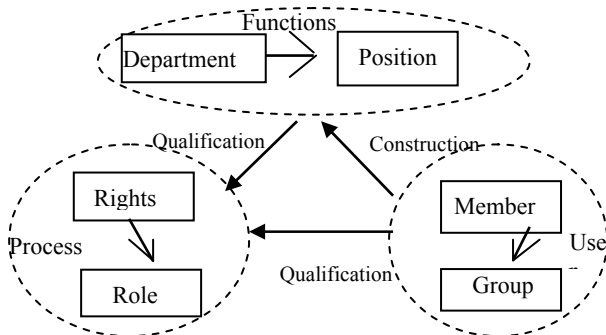


Fig. 3. User management in the workflow design system

The attributes of users can be defined or modified by this tool. The unified user management consists of three functions: organization management, user management and authorization management. The relations among them are illustrated in Fig.3.

*WSL compiler* converts a WSL script into an executable workflow. An executable workflow can be interpreted by workflow engine.

*Workflow engine* is used to execute a workflow service. The users work on a running instance of the workflow service.

*Applications* are the related utilities invoked by the instance of service.

The workflow editor supports interactive editing operations. Users can visually create, edit, save, and convert the specification of a specific workflow service. Users can flexibly manipulate a workflow service, such as creating an activity, deleting an activity, moving an activity and converting a visual design into WSL specification. Authorized service users can take the user management tool to visually edit the user structure for workflow service. Available operations for the user tree are as follows: create, open, close, save a tree or non-leaf node; add, delete, move a node; define, modify, display the attributes of a node.

### 5.2 Workflow Engine

Our system is designed and developed according to the WfMC reference model [1]. The workflow engine is dynamically modeled as a state transition machine. The instance of workflow changes its state in response to external events or under the control of workflow engine.

Fig. 4 illustrates possible state transitions labeled with the conditions. The descriptions of states are as follows: *Initiated* - Workflow process, including the related date and information, is created. But it is not executed if the requirements are not met. *Running* - The instance of workflow is executed. The activities of workflow can be executed if the conditions are met. *Active* - One or many activities in the workflow is executed. *Suspended* - The instance of workflow is suspended. The activities of workflow cannot be executed until the process returns the state *running*. *Completed* - The instance of workflow meets the conditions for finishing. All of operations after completing process are executed, e.g., recording error information, restoring data. Then,

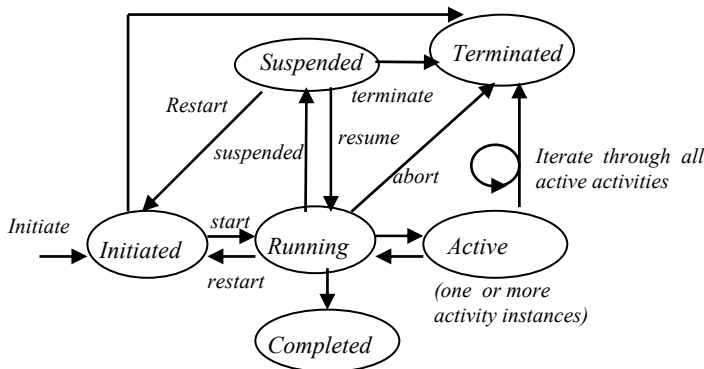


Fig. 4. Dynamic model of workflow engine

the instance of workflow is destroyed. *Terminated* - The instance of workflow is terminated before normally finishing. All of operations after completing process are executed, e.g., recording error information, restoring data. Then, the instance of workflow is destroyed.

During the execution of the workflow instance, once the workflow engine starts an activity, the activity should not be suspended or stopped. This means that workflow process is suspended, restarted and terminated only if all of running activities end. Moreover, it is necessary to treat various activities together as “atomic unit”, and those atomic units must be completely executed once they are executing, otherwise the atomic unit is needed to restart if some exceptions occur.

### 5.3 Implementation Techniques

The workflow system prototype is developed using .net and VC in Windows XP. .net and VC resource library provide a lot of necessary classes in developing the visual system, so it is easy to design and implement a visual interactive development environment. In designing a workflow, the user defines, manages and operates workflow by the tree structure and icon-based visual design. When a workflow service is executed, the system will start workflow engine to handle the information of workflow service and make it active. The workflow management system is responsible for the management and control the whole lifecycle of workflow service.

## 6 Conclusion

This paper proposes a workflow service model based on temporal logic and the philosophy of XYZ system. This model supports the formal specifications for a workflow service, such as the abstraction of activities, synchronization among activities, and the specification of step-wise refinement design procedure of a workflow service. The main advantage of this model is that it can specify the specification of workflow process and its evolution, which are difficult to be supported by the previous models. Using the proposed model, we have designed a workflow specification language and developed an interactive workflow design system prototype. In the next stage, we will further study the models and implementation techniques for the workflow management systems, and develop a powerful interactive workflow design environment.

## Acknowledgement

The work presented in this paper is supported by the National Natural Science Foundation of China (60242002), the National Grand Fundamental Research 973 Program of China (2002cb312200) and the State Key Lab of Computer Science, Institute of Software, Chinese Academy of Sciences. Special thanks to Mr. Hongli Xu, Mr. Xiaodong Huang, and Mr. Yi Duan for technical discussions and software implementations.



## References

1. WfMC. Workflow Management Coalition, <http://www.wfmc.org>.
2. Li, H., Shi, M.: Workflow models and their formal descriptions, *Chinese Journal of Computers*, Science Press, 26(11) (2003) 1456-1463
3. Tang, Z.S., Zhao, C.: A temporal logic language oriented toward software engineering, *Journal of Software*, Science Press, 5(12) (1994) 1-12
4. Tang, Z.S.: *Temporal Logic Programming and Software Engineering (I, II)*, Science Press, Beijing (1999)
5. Ma, H., and Tang, X.: Design and implementation of multimedia authoring language MAL, *Journal of Software*, Science Press, 9(12) (1998) 889-893
6. Ma, H., Liu, S.: Multimedia data modeling based on temporal logic and XYZ system, *Journal of Computer Science and Technology*, Science Press, 14(2) (1999) 188-193
7. Ma, H., and Zhao, C.: Temporal logic based hypermedia specification, *Journal of Computer Aided Design and Graphics*, Science Press, 11(6) (1999)
8. Sadiq, W., and Orłowska, M.: Analyzing process models using graph reduction techniques, *Information Systems*, 25(2) (2000) 117-134
9. Ellis, C., and Nutt, G.J.: Modeling and enactment of workflow system. *Proc of the 14<sup>th</sup> International Conference on Application and Theory of Petri Nets*. Chicago, (1993), 1-16.
10. van der Aalst, W.: The application of Petri nets to workflow management. *Journal of Circuits, Systems and Computers*, 8(1) (1998) 21-66
11. Salimifard, K., Wright, M.: Petri net-based modeling of workflow systems: An overview. *European Journal of Operational Research*, 134(3) (2001) 664-676
12. Endl, R., Knolmayer, G., Pfaher, M.: Modeling processes and workflows by business rules. *Proc of the 1<sup>st</sup> European Workshop on Workflow and Process Management*, Zurich, Switzerland, (1998), 47-56
13. Davulcu, H., Kifer, M., Ramakrishnan, C., Ramakrishnan, I.: Logic based modeling and analysis of workflows, *Proc. ACM Symposium in PODS'98*, Seattle, USA, (1998), 25-33
14. Haake, J., Wang, W.: Flexible support for business process: Extending cooperative hypermedia with process support. *Proc. ACM SIGGroup'97*, Arizona, USA, (1997), 341-350
15. Medina-Mora, R., Winograd, T., et al.: The action workflow approach to workflow management technology, *Proc. ACM CSCW'92*, Toronto, (1992), 281-288

# Research on Cooperative Workflow Management Systems

Lizhen Cui and Haiyang Wang

School of Computer Science and Technology, Shandong University,  
Jinan, 250100, P.R. China  
clz@dareway.com.cn

**Abstract.** Existing workflow management systems assume that each task is executed by a single worker. There is usually no support for group and cooperative work concepts. This paper extends the traditional organizational model with group concept, proposes a cooperative work enabled workflow model, and discusses the implementation of this workflow system. This involves a marriage of workflow systems and some cooperative tools. The model and system are then illustrated through a case study. The results of applications show some advantages of the proposed approach for supporting cooperative work and potential applications in next generation workflow management systems.

## 1 Introduction

Most publications on workflow management focus on process or control-flow perspective, neglecting the representation of organizational structures and the cooperative work [1], as they relate to a workflow management system. Thus, there is a lack of consensus on the type of group structures to be supported. For example, IBM's MQ Series Workflow [2] supports both organizations and roles. Both workers and work items are assigned to roles. A worker may be linked to multiple organizations and an organization may be visible to multiple workers. Another example is Staffware system that supports the concept of a so-called work queue. The available systems have no support for cooperative work. This lack of consensus is also illustrated by the absence of any proposals from the Workflow Management Coalition (WFMC, [3]) concerning the representation of organizational structures and the cooperative work. Although there is a working group on resource modeling (WFMC/WG9), no standards have been proposed. WFMC proposes a workflow reference model, in workflow modeling, defines a process definition meta-model. In this meta-model, workflow definition refers to a hierarchical "role model" to describe organization structure and role information in organization. But it does not provide an organization model that has enough expression ability. So it cannot adapt complex organization structures in the execution of business processes. For instance, today, many activities in business processes have become group activities.

Today's workflow systems assume that each work is executed by a single worker. Even though a work item can be assigned to many workers, from viewpoint of the system, a worker with the proper qualifications selects a work item, executes the

associated work, and reports the result, so the work is executed by a single worker. There is usually no support for group, i.e., groups of people collaborating by jointly executing work items, e.g., the group of software programming development, the program committee of a conference. We think that the reason is that the workflow model is lack of supporting group and cooperative work concepts.

Groupware technology offers support for people working in the group, however, these systems are not equipped to design and enact workflow processes. Systems such as Lotus Domino Workflow [4] provide a marriage between groupware and workflow technologies. Unfortunately, these systems only partially support a group working on a work item. For example, for each work item one needs to appoint a so-called activity owner who is the only person to decide whether an activity is completed or not, i.e., a single person servers as the interface between the workflow engine and the group. Clearly such a solution is not satisfactory. Moreover, there is neither explicit modeling of groups nor any support for people working in the group. The only group-related functionality supported by Domino Workflow is sharing of documents.

The scope of this paper is limited to the modeling of workflow and organizational structures in the context of cooperative support. Current workflow technology is not cognizant of group. This is a major problem since groups are very relevant when executing workflow processes. Consider for example the selection committee of a contest, the steering committee of an IT project, and the board of directors of a car manufacturer. In addition to providing explicit support for modeling groups, it is also important to recognize that individuals typically perform different roles within different groups. For example, a full professor can be the secretary of the selection committee for a new dean, and the head of the selection committee for tenure track positions. These examples show that modeling of group should be supported by the future generation of workflow products. In this paper, we explore concepts and technologies for making workflow management systems cooperative work enabled.

The remainder of this paper is organized as follows. First we introduce the group concept and extend the workflow processes definition meta-model to incorporate support for cooperative work. In section 3, we discuss possible realizations using groupware technology. In section 4 we give a case study. Section 5 concludes this paper.

## **2 Workflow Model Supporting Cooperative Work**

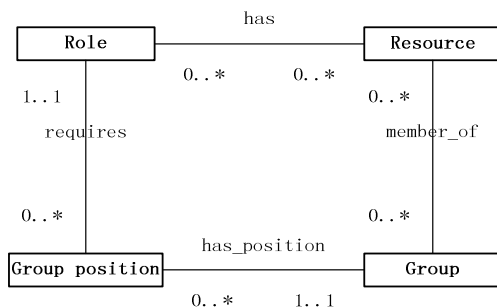
In this section, we introduce the workflow model supporting cooperative work. First, we discuss the group concept. Then, we introduce the meta-model and its class object in detail. Finally, we illustrate this model how to support cooperative work in definition phase and execution phase.

### **2.1 The Group Concept**

In existing workflow systems, work items are distributed over resources. Although a work item may be offered to man resources, from the perspective of the workflow management system, a work item is still executed by one resource. To decouple the workflow process definition from concrete resources, the concept of role was

introduced as explained in the standard of WFMC [5]. To be able to model workflow supporting cooperative work, we introduce some correlative group definition.

A group is a set of resources. A group can have several members and one person can be a member of many groups. Some groups are created on-the-fly, i.e., the group is created the moment an activity requires a group of a specific type. Other groups are of a more permanent nature and handle many activities. Fig.1 shows the group model, which extend the role model of WFMC.



**Fig. 1.** The Group Model

A group position is a specified role within a group. For example, consider a policy that “the chair of the selection committee should be a full professor, while other members should be full-time faculty of any rank.” In this example, the chair and member have different roles within the group. The association “requires” links each role to a group position. Association “has” links roles to resources. Every resource can has several roles in group, and every role can be hold by several resources. Association “has position” links group and group position. Note that every group has several group positions. The association “member of” shows that a group can have a set of resource, and every resource can be a member of several groups.

We have extended traditional role model of WFMC. In the following section, we propose a workflow meta-model, which can well support cooperative work.

## 2.2 Workflow Meta-Model Supporting Cooperative Work

Meta-model is a model defines a language for expressing a model [6]. Workflow meta-model describes all elements in workflow, relationship of these elements and attribution of them. WFMC has developed a basic process definition meta-model [5], which use role to specify the mapping of activities onto members. Generally, a role concept is used to decouple the workflow process definition from concrete members, e.g., activity approve a loan should be executed by Mr.Zhang should be avoided. Therefore each activity is assigned to a role, e.g., activity approve a loan should be executed by someone with the role manager. But the role concept in this model has no uniform definition. In order to support cooperative work, we introduce group concept and other relative concepts into basic meta-model and propose a meta-model supported cooperative work, which is illustrated in Fig. 2.

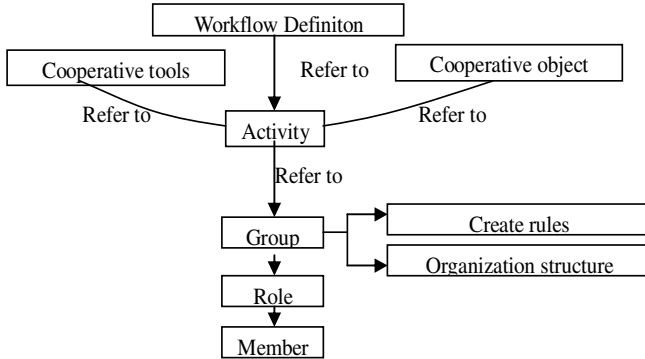


Fig. 2. Workflow meta-model

- Activity: mainly is cooperative activity, e.g., software development evaluation, group decision in a company.
- Group: a number of people organized together, e.g., expert group of evaluation, program committee of a conference. A group can have several members and a person can be a member of many groups. Its model is illustrated in above section.
- Cooperative object: the sharing resources accessed by group members in cooperative work, e.g., patient examination result information in consultation.
- Create rules: some groups are more permanent and handle many activities, others are created temporarily, i.e., group is created at the moment that an activity requires this type of group. In any case, some rules and requirement are needed when the group is created.
- Organization structure: composing and construct of the group, it can reflect group composing from many point of view, e.g., group department organization structure represents hierarchy of members in group, for example, a company have a board chairman, a general manager, department managers from top to down. Group business organization structure represents composing of role function, is oriented business processes, for instance, a software company have a software development evaluation group, project no.1 development group, project no.2 development group.
- Role: besides common used roles in business processes, there are some special roles of cooperative work, e.g., presider, coordinator.

This workflow meta-model can support cooperative work in two sides: (1) it can describe group and group work in definition phase; (2) it can offer support of cooperative work in execution phase, including group activity assignment, group creating, invoking cooperative tools, and so on.

### 2.3 Support of Cooperative Work in Workflow Definition

This workflow model refers to three new elements: group, cooperative object and cooperative tools. In traditional process model, role is used to specify the mapping of activities onto members. Now group replaces it, and can extensively support

cooperative work. Members of group do not swarm together, but according to some structures. So organization structure is an important attribution of group object, and is changeable when business requirements have been changed. All organization structures can be stored in database. We can retrieve them to know about group structures from several sides. Members of group have roles or positions, e.g., manager, president. If a group has only one role, it can be seen a special group. Roles are included in group, and can support cooperative work as well as traditional process definition.

Cooperative object can be transferred and received between members of group. Cooperative tools are some cooperative application tools outside workflow system. It can be invoked by workflow engine, and have strong ability for communication of members.

## 2.4 Support of Cooperative Work in Workflow Execution

In workflow execution, especially cooperative work, some cooperative objects must be created and stored in database, which can be accessed by any group member related this cooperative activity. The execution of cooperative activity may be outside of workflow system, so workflow engine can invoke some cooperative tools to transfer cooperative objects from one member to the others. Following, we introduce some cooperative behaviors.

- Group creation: here, we just discuss group creation temporary when a cooperative activity is executed. Workflow services can broadcast all persons have proper qualifications to create group. For example, if an activity requires a group that must have one professor and two associate professors, workflow engine notify this activity's information to all professors and all associate professors. Once a professor accept this activity, which automatically withdraw from other professors. Similarly, two associate professors composing can use the same method. The shortcoming of this method is that the members in a group may be not harmonious. We can use a more natural method, which is a person is assigned to be a coordinator whose task is to create this group. Finally, it submits to workflow engine the group member list, and then workflow engine notifies to them and makes confirmation.
- Processing cooperative objects: cooperative objects are sharing for all related group members, and can be transferred from one to the other. For keeping consistency, workflow engine must record the receiver of this cooperative object and member who current accesses or modifies this cooperative object.
- Completion of cooperative activity: there are two methods to decide the completion of cooperative activity. First, each group member would inform the workflow system separately. Second, a group coordinator, who is elected or appointed, decides the completion of cooperative activity.
- Interaction of group members: an important function is the interaction ways of group members. All group members may work at the same time at different place, or coordinate with each other using email or electronic discuss board. So some cooperative tools can be as the interaction ways of group members.

### 3 The Implement of Cooperative Work Enabled Workflow Systems

Based on workflow model supporting cooperative work, we want to use some mature workflow technologies and cooperative technologies, but not design a brand new workflow system, to construct a cooperative work enabled workflow systems. Our architecture of cooperative work enabled workflow system is illustrated in Fig. 3.

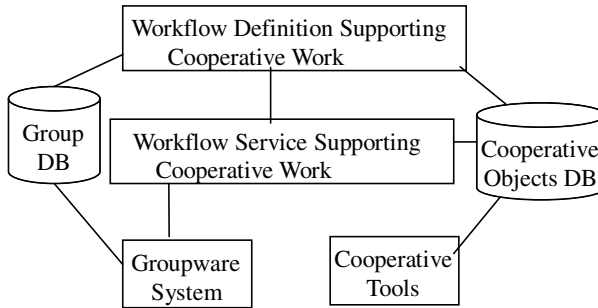


Fig. 3. Architecture

**Workflow definition supporting cooperative work:** we can extend traditional workflow definition tool, make it support cooperative work definition based on above workflow model. Workflow definition may refer to some group information defined by groupware system and some cooperative objects.

**Group database:** stores some group information used in groupware systems or defined by workflow definition tool.

**Cooperative objects database:** stores some resources information used by group members during cooperative activities.

**Workflow service supporting cooperative work:** can decide the start of cooperative activity, assign the cooperative activity to a special group, invoke some cooperative tools, and transfer cooperative objects information to other group members. Workflow engine can call on some groupware system to arrange complex cooperative activities. Groupware system also can invoke cooperative tools. Finally, it submits the cooperative activities execution results to workflow engine.

**Groupware system:** can support some cooperative activities. Even though it has various standard models such as a calendar for scheduling, e-mail support, and video support, in addition which can keep track of availability of meeting rooms, its main functions include:

- Access sharing objects.
- Communication between participants of group. Typical examples are electronic mail systems and video-conference systems, and basic issues that need to be addressed are message passing, communication protocols, and conversation management.

- Functions supporting cooperative behaviors. Such as function of auto co-design and so on.

Cooperative tools: mainly include real-time cooperative tools and web-based cooperative tools. Real-time cooperative tools are a blend of electronic whiteboard, Internet chat and video-conference software. The collaboration process may involve synchronous communication through video-conference, application sharing and data sharing or instant messaging services (e.g., Microsoft's NetMeeting).

Group work enabled workflow systems should provide following interfaces supporting cooperative work, which integrate existing workflow systems, cooperative tools, groupware systems and workflow definition tools:

- Interfaces between workflow process definition tools and workflow engine should add some concepts supporting group and cooperative work. For example, it can adopt group definition standard used in existing groupware systems. In this way, workflow process definition tools can access whole group information from group systems, and workflow engine also can interpret directly this definition according to standard.
- The main functions of Interface between workflow engine and groupware system include:
  1. Interpret workflow definition supporting cooperative work.
  2. Call on group systems to create group.
  3. Assign cooperative activity to groupware system. Workflow engine call on groupware system to execute cooperative activity and provide details such as: group members information, deadline for the completion of this cooperative activity, cooperative objects information, interaction ways of members, group organization structure, based on above, groupware systems arrange members to complete this cooperative activities. At the end of this activity, groupware system notifies workflow engine that this activity has been completed, and returns various result information of active activity instance to workflow engine.
- Interface between workflow engine and cooperative tools. Its primary function is that workflow engine can invoke some cooperative tools when cooperative activities need them to help.

## 4 A Case Study

There are many activities that involved considerable group operation. A conventional workflow system would not be very efficient in such situations. To illustrate how the proposed model and systems handle this situation, we consider an example of healthcare collaboration.

A physician in a remote clinic with limited medical resources is the user of our workflow system. Sometimes, a patient's condition is more complicated and the physician can not handle it within the clinic with its limited resources or the physician does not have enough experience with the patient's condition. Hence, he probably has to consult it with one or several physicians in other health centers with advanced high-tech healthcare facilities. In order to do consultation, patient information is



bundled as a collaboration object and stored in a web accessible repository, and the collaboration partner accesses this patient information to provide his professional comments by using our cooperative workflow system. Fig. 4 briefly shows the execution of the healthcare collaboration scenario based on our workflow model and system supporting cooperative work.

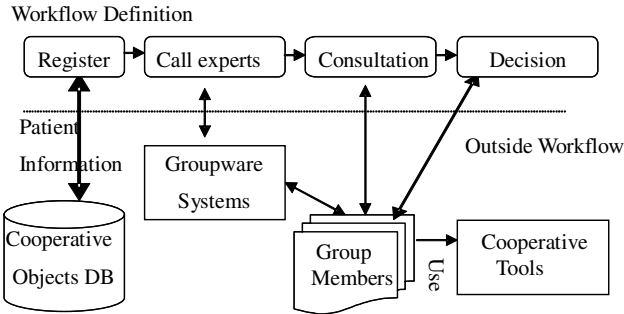


Fig. 4. A Case Study

As shown in Fig. 4, the whole process and each activity are designed and controlled by cooperative workflow engine. In this workflow process, consultation and decision are typical cooperative activities. After registration, patient information, as a cooperative object, is stored in a web cooperative objects database. Groupware systems are used to create group, coordinate group members and other relative cooperative operations in workflow execution, which are outside of workflow engine scope. Group members are controlled by groupware system to cooperative work each other. But each group members' cooperative work is still scheduled by workflow engine. Each member can work cooperatively, such as holding a consultation to decide whether an operation was necessary, through groupware system. In addition, group members can interact with each other using some cooperative tools.

## 5 Conclusion

This paper starts with the discussion that group and cooperative work supports are missing in current workflow management systems. Yet, there are plenty, real-world applications where tasks are performed by groups, so research on workflow management systems supporting cooperative work is very important. Even though there are many research results in the field of groupware systems and CSCW (computer supported cooperative work), few researches have been found on cooperative workflow management systems.

We extend the traditional role model of WFMC's standards using group model, and propose the workflow model supporting cooperative work, explicitly introduce that how to support cooperative work, construct a cooperative work enabled workflow system, and briefly discuss the implementation of this system. In addition, a case

study is introduced to illustrate the execution of this system. In the future, we plan to apply some concepts of this paper to our current project of digital healthcare integration platform.

## References

1. zur Muhlen, M.: Resource Modeling in Workflow Applications. Proceedings of the 1999 Workflow Management Conference, Muenster, Germany, November, (1999)137-153
2. IBM.: IBM MQSeries Workflow Concepts and Architecture, Technical Report. International Business Machine co., July, (1998)
3. Lawrence, P. (ed.): Workflow Handbook 1997. Workflow Management Coalition, Wiley, New York (1997)
4. Nielsen, S.P., Easthope, C., Gosselink, P., Gutsze, K., Roele, J.: Using Lotus Domino Work- flow 2.0, Redbook SG24-5963-00. IBM, Poughkeepsie, USA (2000)
5. Workflow Management Coalition: Workflow Reference Model. Workflow Management Coalition Standard, WfMC-TC-1003 (1994)
6. Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modeling Language Reference Manual. Addison Wesley Longman, Inc. (1999)
7. van der Aalst, W.M.P., Kumar, A.: Team-Enabled Workflow Management Systems. Data and Knowledge Engineering, 38(3) (2001)335-363
8. Nunamaker, J., Dennis, A., Valacich, J., Vogel, D., George, J.: Electronic meeting systems to support group work, Communications of the ACM, 34(7) (1991) 40-61
9. Sheth, A., Kochut, K.: Workflow Applications to Research Agenda: Scalable and Dynamic Work Coordination and Collaboration System. Proceedings of the NATO Advanced Study Institute on Workflow Management Systems and Interoperability, (1997) 12-21

# Effective Elements of Integrated Software Development Process Supported Platform

Min Fang, Jing Ying, and Minghui Wu

College of Computer Science, Zhejiang University,  
Hangzhou, P.R. China, 310027  
fm@zju.edu.cn

**Abstract.** Modern software development puts much emphasis on unified and standard software development processes, such as RUP (Rational Unified Process), TSP (Team Software Process), PSP (Personal Software Process), and XP (Extreme Programming). In order to support these standard processes, this paper proposes a software development process supported platform that takes workflow engine as its core and contains a number of effective elements such configuration management, knowledge management, and agent-assisted personal software process. Based on this integrated platform, the development process of software organizations can be fully automatically controlled with high performance.

## 1 Introduction

Since 1990's, unified and standard software development processes have drawn more and more attention in the software development industry. A typical example is the emergence of software development processes such as RUP (Rational Unified Process), TSP (Team Software Process), PSP (Personal Software Process), and XP (Extreme Programming). Under the circumstances of emphasizing processes in software development, the performance of processes, if implemented only according to rules and regulations, will not only lower the performance efficiency, but also greatly reduce the validity of workflow. In this case, an automated development supported platform with strict workflow control is desperately needed.

Workflow management system is a technology that has developed quickly in recent years and widely applied in commerce, service and other industries. The technology has achieved streamline organization workflow of high efficiency [3]. The application of workflow technology has an overwhelming advantage in IT industry, which is oriented to information process. That is, information routine may be better realized on the basis of workflow routine, the manifestation of which in software development support is the accomplishment of integrated configuration management.

In order to support the standard software development processes, this paper proposes a software development process supported platform that takes workflow engine as its core and contains a number of effective elements such configuration management, knowledge management, and agent-assisted personal software process. Based on this integrated platform, the development process of software organizations can be fully automatically controlled with high performance.

## 2 Platform Framework

### 2.1 WSDPP Framework

The current software development process supported platform provides some functions as follows: version control, bug track, project management, mailing list, and membership configuration (Figure 1). Those functions satisfy the minimum subset of the basic needs in software developments. However, those functions are scattered and are not associated in a good way. An intellectual and effective core is needed.

Project Workspace		
Software Development	Knowledge Management	Technical Communication
Version control Issue tracking IDE integration	Knowledge archive Document and file management	Mailing lists Discussion Forums News
Project Administration		
Membership Project configuration Reporting		
Security & Permissions		
Security Role-based permissions		

**Fig. 1.** Common platform framework

After the introduction of workflow and agent technologies, the proposed WSDPP (Workflow-based Software Development Process Platform) has overcome the problem of cohesion of the general supported platforms.

WSDPP has realized the automatic controlling of processes with the workflow engine at the core. After the introduction of the process template based on RUP/TSP/PSP, and the appropriate reduced version, the standard development processes of the organization, team and individual can soon be established. At the same time, assistant process such as review track, knowledge management, and technical service support can also be realized. PSAF process meta model, which is based on the improvement of FlowMark and EPC meta model [1,2], combines closely process, system module information, artifact, electronic form and organization, and ties the configuration management with the process management simultaneity. In WSDPP, a relational database management system provides the process definition and the process instance with a permanent storeroom. The correspondent model of organization, artifact and electronic form are also stored in the relational database [9].

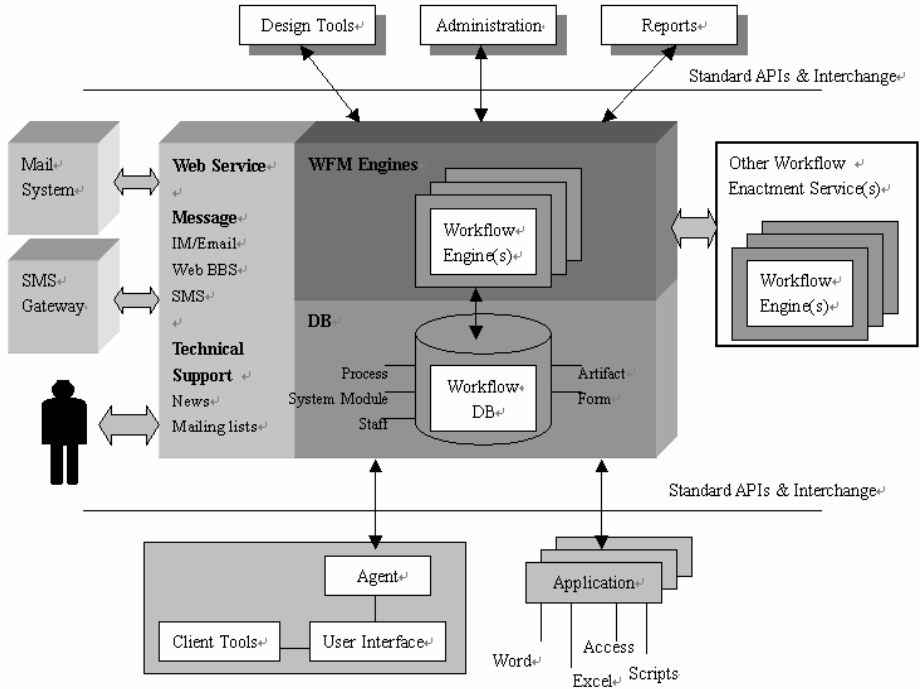


Fig. 2. Intelligent, integrated platform framework

The process definition is based on XPDL (XML Processing Description Language), and is described using UML, which is comparatively easy for both developers and users. From UML to XPDL and further to Relational Database implementation, there are two unilateral translation programs that are important to realize the third class switch [5,6,7].

The platform provides the modules of management and report forms for the implementation of workflow and performance analysis. In the interaction with other workflow engines, the general Web Service is adopted. Platform also helps to realize the accessorial connection of platform and developers through external Email system and the SMS (Short Message Service) gateway [5].

## 2.2 PSAF Meta Model

A lot of sensible process models have been put up since the development of the workflow technology. But it is far from satisfaction as for the adaptability of the models [4]. Most models of workflow start with the description of the process such as Petri net, state graph, and activity network. Those kinds of models can reflect the order of the process directly, but they cannot be applied in dealing with the complex process logic. They also fail in providing enough modeling concepts and displaying the model with a lot of restricted elements.

In the platform, PSAF (Process-System module-Artifact-Form) process Meta model (Figure 4) is formed base on the expansion of IBM FlowMark and EPC by Keller [1, 2].

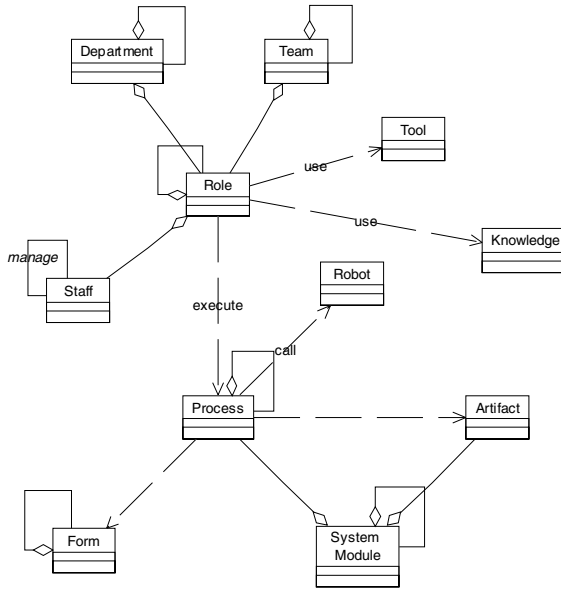


Fig. 3. Static structure view of workflow elements

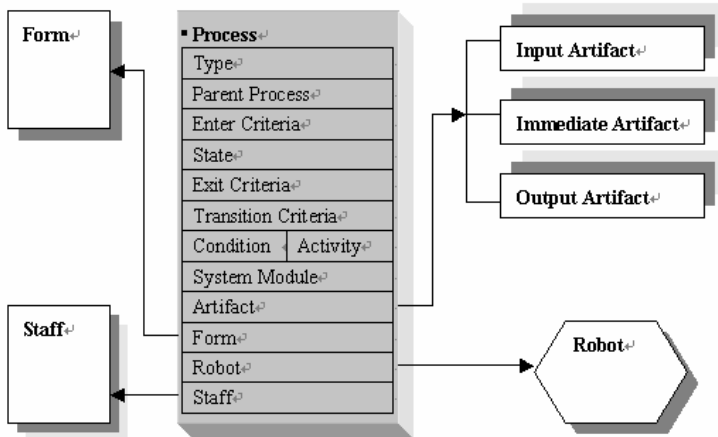


Fig. 4. PSAF Meta model

PSAF Meta model has realized the relationship between the elements in Figure 3: the process is linked with the system module, artifact, electronic form, staff and

automatic machine Robot [11]. In every process, there are enter, exit and transition criteria. And the transition criteria can be described as a table with conditions and activities. When the exit criterion is satisfied, the workflow engine will review all terms in the transition criteria table and activate the correspondent activities if the condition is satisfied.

The conditions in enter, exit and transition criteria are written in a normal script language such as Perl, PHP, VBScript. Through the script language, we can approach all the object properties in Figure 3. In that way, the relationship of order dependence can be enriched and expanded, and the controlling of the activities is also reinforced.

When the type property is “auto”, the Robot property of Meta model PSAF is in effect. The property of Robot can also be defined by script language, and all the activities, such as the printing of the report form by using the background batch program, can be realized automatically by using the external application program.

As for the process modeling of the big and complex system, the difficulty lies in how to deal with the complexity. The most effective way is to do in a top-down approach, and make them accurate step by step. PSAF Meta model is in favor of the multi-level activity models. When the active Parent Process property is not in blank, the activity should be marked as nesting.

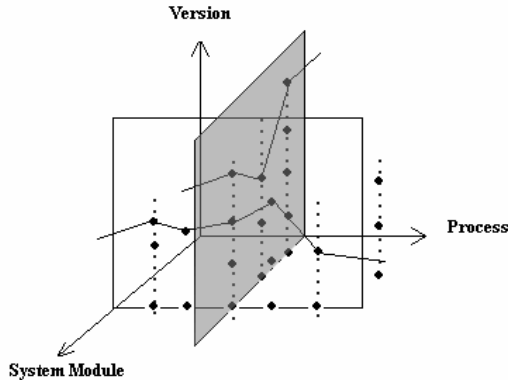
## 3 Support for RUP

### 3.1 Integrated Configuration Management

The present tools for configuration management can usually provide storage bank of catalogue model for different versions, so different storage catalogs are used to distinguish development process from corresponding system module in different versions of artifacts generated by the development process. There is no direct connection between an artifact and a process. The developer can put use case specifications either in the implementation catalog or in the maintenance catalog. Furthermore, only if rights are allowed, can the developers who do not take part in the exploration of artifact also gain all the artifacts.

The confusion is ultimately resulted from separation of configuration management and workflow. In WSDPP, artifacts and the processes are closely connected. An artifact can only be touched in the processes, that is to say, only those who create, change the artifact and who are in charge of the next processes on the basis of the artifact can get, check out and check in the artifact.

Under catalog module-based configuration environment, only one-dimensional attributes are provided to mark artifacts. Artifacts with more attributes can be produced only by inserting other dimensional attributes into one-dimensional attributes, which are often not rigid and cannot be understood directly if operated manually. While in WSDPP, artifacts are endowed with multi-dimensional attributes, among which Process, System Module and Version are the three most important attributes. The three attributes, used to mark the production processes of the artifact, system modules it is affiliated with and its version, consist of three-dimensional coordinate description of the artifact (Figure 5).



**Fig. 5.** 3-Dimensions of artifact

It is worth mentioning that check out and check in of artifacts in one workflow are visible only to the developers themselves. Only after an artifact is delivered, reviewed and labeled, can the artifact be opened to other developers.

When any change should be made on the artifact that has been delivered, reviewed and based, developer must create a new change process. The change of workflow will be rejected if the reasons of change are not adequate, whereas the original developer would alter it. After the change process begins, the reasons of its change and possible influential system module are presented and then the changed artifact is delivered. Then the review process begins automatically. As the reexamination process ends, all the workflows in the current system module and all the versions of the artifact would become “disable” and the passive workflow is automatically stimulated. So far, the whole change process ends.

### 3.2 Knowledge Management

In traditional Manufacturing Information Systems (MIS) and Enterprise Resource Planning (ERP) systems, KM (Knowledge Management) is usually provided as a separate module. While in the supported platform, KM is only a common workflow. Since a developer’s knowledge is described in a formal knowledge file, KM workflow is established. Then the developer fills the knowledge attributes form, delivers the file. Finally an assessor reviews the file and determines the attributes of knowledge.

After the knowledge file is assessed, it will appear in the help link of the corresponding workflow. Knowledge file can also be obtained through search of knowledge bank. Everyone who reviews the file can rank the knowledge.

## 4 Support for TSP

### 4.1 Environment Workflow

There are often some similarities between different projects in a software development organization. The process is similar between different projects and some



guidelines are probably identical. Platform provides some standard development workflow templates such as RUP, XP for the environment team. Therefore, an environment team can build organization-wide process through workflow templates just as writing a document through a word template. The organization can benefit significantly by using this approach.

## 4.2 Project Plan

When the instance of defined process is put into use, the workflow engine creates a plan process combined with real-time process definition, system model, human resources available, and BDI attributes of the developers' agents.

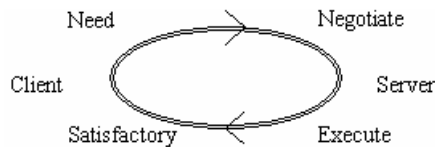


Fig. 6. Speech Act model

Then, based on the speech act model (Figure 6), the developer will negotiate with the project manager about workflow and finally decide whether he will accept the assignment, which sometimes may be competed for among several developers. If he accepts it, he may put forward his own opinions with his personal conditions, and formally execute the process after all the conditions of the task reach full agreement.

## 4.3 Review and Change Workflow

Review, which is obviously effective as a filter for flaws, is often inserted in two contiguous development processes such as in design and implementation. For the sake of higher quality of products, review of all the processes is preferred, but which will make it become the bottleneck of the whole development process or result in its low filtering rate owing to the huge workload of QA (Quality Assurance). In this case, the tradeoff is to review important processes directed towards key system modules.

Following one complete workflow, review will automatically begin or not do so according to its system setting. If it begins, assessor examines the artifact delivered, its Enter criteria is whether the artifact is basically qualified. The review will not be performed until the artifact is qualified. Every bug found in the review will be traced. The Exit criteria in review can be performed if only all the bugs found are resolved or cancelled. Until now, the review workflow is over and begins the next development workflow.

An example of WSDPP is presented in Fig.7. Project manager can make a detailed plan on the left page in the figure. The plan has a set of attributes, such as process, iteration, developer, development start/end time, reviewer, and review start/end time. When an artifact has been developed, according to the plan, review process is automatically started. Reviewer will receive an Email notifying him some artifact is

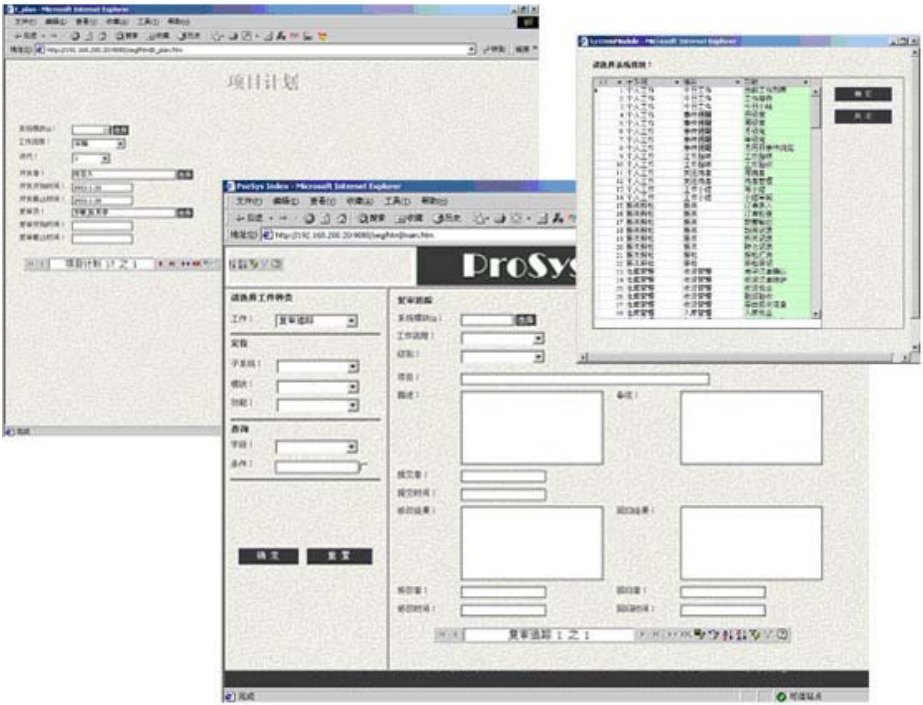


Fig. 7. Example of WSDPP

finished. After the reviewer has checked the artifact, he should input examination results on the middle page, and then the developer can take some remedies with the help of the reviewer's comments. In all of these processes, system module is a key element, and every plan or review result has the system module attribute.

#### 4.4 Communication

For a software organization, knowledge of employees is their most valuable treasure which can be released in a way similar to the equation  $E=mc^2$  by communication among employees. Therefore, an open and effective communication environment is indispensable and forms of communication should be versatile.

The integration of IM (Instant Message) instrument, Email, Web BBS is realized in this platform to facilitate the instant communication and discussion of news as well as technologies among developers.

### 5 Agent-Assisted PSP

In the standard workflow reference model defined by WfMC, user is part of the workflow model. But in the process of realization, the present workflow management systems often regard the individuals involved in the workflow as kind of "machines".

That is, after the process and plan are ascertained, the processes are directly assigned to specific individuals and required to be completed under the assigned conditions, which takes into consideration little of individuals' willingness and capability of accepting the task. Thus, it brings about many problems in the application of workflow, such as exceptions in the workflow implementation [10].

In WSDPP, speech act based workflow model is partially adopted to denote the behavior of commission and assumption between server and client. Furthermore, the server is a developer-based personal Agent.

### **5.1 Interface of Agent and Workflow Engine**

The interface of agent and workflow engine is laid beside the workflow engine to simplify the process, and this interface will function as task category location, knowledge database location, agent-set BDI (Belief, Desire, Intention) model [8], etc. A developer first should set personal agent BDI model, input personal desires, intentions, and knowledge (the attributes of which should be confirmed by assessors). Personal desires can be interpreted as whether individuals are willing to accept the maximum tasks under acceptable conditions, or whether they will try their best to reduce task assignments on the premise of accomplishing the least workload.

Agent plays several roles in the process of PSP execution. For example, it may provide relevant knowledge and script direction in each stage; it can prevent IM and email messages when the developer do not like to be bothered, and remind him of executing certain tasks at fixed time by automatically answer messages according to the message-action list; it may give some suggestions about PSP based on organization productivity database; it also can automatically extract relevant content from knowledge base and BBS.

### **5.2 Data Collecting in PSP**

In the stage of execution, the developer will bring tasks into personal PSP and produce personal tasks. Through PSP tools provided by the client platform, he will collect all kinds of process time, scale, quality data, and automatically make process summary report. In addition, personal productivity database will be updated, and concluded in organization productivity database in a certain interval.

## **6 Conclusion and Future Work**

This paper introduces the structure of a workflow-based software development process support platform and an improved PSAF process meta-model. This platform also provides agent-based assistance in interfaces defined by workflow reference model. Like other workflow management systems, this platform has similar difficulties, such as workflow change, refusal, and workflow definition reversed by workflow implementation and exception handle. This demands us to focus our attention on extracting, defining and handling the hidden relationships among all the elements inside work artifacts, UML workflow definition, XPDL interchange, and the improvement of BDI structure of personal agents.

It is believed that the proposed workflow-based software development process supported platform will turn out to be a natural and effective “bond” between humans and workflow systems.

## References

1. Mohan, C., et al.: Exotica: A project on advanced transaction management and workflow systems. *ACM SIGOIS Bulletin*, 16(1) (1995) 45--50
2. Dehnert, J.: Four Systematic Steps Towards Sound Business Process Models, *Proceedings of the 2nd International Colloquium on Petri Net Technologies for Modeling Communication Based Systems*, Berlin, (2001) 55-64
3. Shi, M., Yang, G., Xiang, Y., Wu, S.: WfMS: Workflow Management System. *Chinese Journal of Computers*, 22(3) (1999) 325-334
4. Fan, Y.: *Fundamentals of Workflow Management Technology*. Springer, Berlin, (2001)
5. Workflow Management Coalition: The Workflow Reference Model. WfMC TC00-1003, (1994)
6. Workflow Management Coalition: Workflow Standard – Interoperability Abstract Specification, v1.0, (1996)
7. Workflow Management Coalition: Workflow Management Coalition Specification: Terminology & Glossary. WfMC-TC-1011, (1996)
8. Rao, A., Georgeff, M.: BDI Agents from Theory to Practice, Technical Note 56, AAIL, (1995)
9. He, Q.: RELATION-BASED Lightweight Workflow Engine. *Jisuanji Yanjiu yu Fazhan, China*. 38(2) (2001)129 (in Chinese)
10. Cai, T., Gloor, P., Nog, S.: DARTFLOW: A Workflow Management System on the Web Using Transportable Agents. Technical report, Dept of Computer Science, Dartmouth College, (1997). <http://www.cs.dartmouth.edu/reports/authors/Cai,Ting.html>
11. Manolescu, D.-A., Johnson, R.E.: Patterns of Workflow Management Facility. <http://www.uiuc.edu/ph/www/manolesc/Workflow/PWFMF/>

# Hierarchical Timed Colored Petri Nets Based Product Development Process Modeling

Hong-Zhong Huang<sup>1,2</sup> and Xu Zu<sup>3</sup>

<sup>1</sup> Department of Mechanical Engineering, Heilongjiang Institute of Science and Technology, Harbin 150027, China

<sup>2</sup> School of Mechatronics Engn, University of Electronic Science and Technology of China, 610054 Chengdu, Sichuan, P.R. China

<sup>3</sup> School of Mechanical Engineering, Dalian University of Technology, 116023, Dalian, Liaoning, P.R. China  
hzhuang@uestc.edu.cn

**Abstract.** Product development (PD) process modeling has been a critical problem in modern PD process management. First we discuss the many benefits of PD process modeling, and then we outline the characteristics of PD pattern in order to provide a full and exact description of it. A powerful modeling language is introduced which is based on the characteristics of modern PD and Hierarchical Timed Colored Petri Nets (HTCPN). A pump development process model based on HTCPN is proposed. The performance analysis of HTCPN and the HTCPN-based process model are presented. Various performance measures of PD process can be generated from the proposed model.

## 1 Introduction

Today's product development (PD) occurs in a highly challenging environment, and companies are under increasing pressure to sustain their competitive advantage by reducing PD time and cost without sacrificing quality. Nowadays, however, engineering products have become more and more complex in response to consumers' requirements for product safety and function. The increasing complexity of the PD and design process calls for concurrent strategy and thus results in large interdependent task groups. The large size of interdependent task groups usually makes team organization difficult and, thus, delays the project's completion. Moreover, as complexity increases, managing the interactions among tasks and people becomes more difficult; it may be impossible to predict the impact of even a single design decision in the PD process. That is why improving the effectiveness of PD is crucial in shortening PD time and lowering costs. As well, developing an effective PD process model to describe and analyze the actions in PD is an effective way to organize processes and resources, minimize unnecessary process iterations, and speed up the PD cycle.

There are several existing process modeling methodologies and tools [1]. Petri Nets, especially Colored Petri Nets have proved to be a favorite modeling language in PD process modeling [2-7], however, research in this field still leaves much to be desired because today's PD processes differ significantly from other business processes. A PD process model can not be viewed as a static representation of an engineering process, but should be a dynamic one since unpredictable changes or exceptions may occur at any time. As a process model to support PD, it should be: (1) a

mechanism to depict the characteristics of PD processes and the dynamics of executing the static process model; (2) an analytical mechanism for analyzing and understanding the important dynamics of the process model's performance.

We developed a powerful modeling language, Hierarchical Timed Colored Petri Nets (HTCPN), to describe the dynamics of engineering processes in light of the typical characteristics of engineering processes. HTCPN has been derived from classic Colored Petri Nets [8] and used to capture and analyze the properties of the collaborative engineering design process. The goal of this work is to create a general process model that can realistically represent the nature of a complex modern design project and analyze the dynamics of its processes. This model can be used throughout the PD process to improve PD effectiveness, predict potential exceptional processes, accelerate communication among people, and guide project management efforts.

The rest of this paper is organized as follows. Section 2 covers a brief introduction to some advantages of PD process modeling, and Section 3 discusses in detail the characteristics of PD; these provide a foundation of modeling. In Section 4 we introduce some dependencies between tasks in modern PD. A formal definition of HTCPN is brought forward in Section 5, and a typical PD process model based on HTCPN is presented in Section 6. The dynamic properties and performance of the model are expatiated on subsequently. Finally, a conclusion is presented in Section 8.

## 2 The Need for PD Process Models

A complex PD project involves a large number of tasks executed by professionals from various disciplines. As complexity increases, it becomes more difficult to manage the PD process and its interactions among tasks and people; it may be impossible to predict the impact of even a single design decision or the omission of a task somewhere in the PD process. So PD process modeling is not only necessary but well worth the effort. There are many reasons for undertaking PD process modeling. The following constitutes the best arguments for doing so in an effective manner [9]:

**Understanding and learning:** A process model helps people to get an overview, to understand what roles they play in the project, and to see who is doing what and when. A transparent process model also supports communication among PD project participants. Good communication makes it easy for people to understand complex processes; it also provides an excellent learning aid for employees who are new or have changed jobs.

**Control and operational management:** In the course of PD, a conflict between different tasks may occur. A consistent process model promotes better communication and conflict resolution.

**Strategic management and planning:** By providing a transparent view and an aid to good coordination, detailed planning and easier management of the actual PD can be achieved.

**Process improvement and shorter PD cycles:** A process model can be used to conduct process performance analyses and to improve the process through rearrangement or reengineering. Moreover, process modeling can shorten PD time.

### 3 Characteristics of PD

In order to adequately describe processes in the PD context, a better understanding of the specific characteristics of this kind of process is necessary. PD process is a series of technological and management activities which organize the creation of a product from definition to production. For today's PD pattern, such as concurrent engineering and CSCW, processes are highly interconnected and collaborated, including feedback-loops and interactions at different hierarchical levels. There are many distinct characteristics of the modern PD process. These characteristics (listed below) are contained in our PD model.

(1) Design phase: PD can be divided into many design phases, such as product planning, conceptual design, detailed design and technical design. (2) Hierarchy: Just as a product always involves subsystems, components etc., corresponding PD tasks to be carried out fall into categories and hierarchies. (3) Concurrency: Macroscopically PD is performed in sequence; however, many microcosmic tasks can be executed simultaneously in some phases. Moreover, if some tasks can be broken down into several concurrently executable subtasks, over-all PD time may be reduced. (4) Iteration: Iteration is the repetition of tasks to improve an evolving PD process. It is a fundamental characteristic of PD processes. (5) Design check: Checking is one of the most important activities in the PD process. It can determine whether the design work should be transferred to the next step, reiterated or redesigned. (6) Prerelease (Overlapping): Overlapping has been described as a core technique for saving PD time in concurrent engineering. It is generally acknowledged that a proper overlapping strategy may save time. Because prerelease can reduce the waiting time of downstream activities, it can greatly shorten the overall PD time.

## 4 Task Decomposition and Task Dependencies

In concurrent engineering, PD tasks are arranged concurrently as much as possible in order to reduce PD time. The PD should be divided into many manageable tasks for process modeling. The different dependencies between these tasks will require different methods of execution.

### 4.1 Task Decomposition

In PD there are not only many independent (uncoupled) tasks, dependent (decoupled) tasks, but also interdependent (coupled) tasks. Dependent tasks can be decomposed into some independent subtasks. Thus, in our PD process, independent and dependent tasks can be executed in series or in parallel. To accomplish interdependent tasks, usually multifunctional teams are organized with team members from different functional departments interacting in every phase of the PD tasks. Unfortunately, the large size of interdependent task groups usually makes team organization difficult and thus delays a project's completion. This necessitates an effective model for decomposing large interdependent task groups into manageable sub-groups. In this paper, we presume that all of the tasks have been decomposed, and can be arranged in some way (series, parallel, overlapping etc), although there will be some tasks which are tightly

coupled and so can't be decomposed into sub tasks. The greatest difference between decomposed and undecomposed is that the latter can be executed only by multifunctional teams.

### 4.2 Task Dependencies

A clear description of the relationships between tasks is a key part of PD process modeling due to the complex inter-dependent nature of design tasks. In this paper, task dependencies are divided into two types, temporal dependencies and resource dependencies. Temporal dependencies establish the execution order of tasks. Resource interdependencies are complementary to and independent of temporal ones and deal with resource distribution among tasks.

In concurrent engineering, there are four primary types of temporal interdependency: series, parallel, prerelease (overlap), and iteration [2]. The dependencies dealt with in this paper are described in Figure 1.

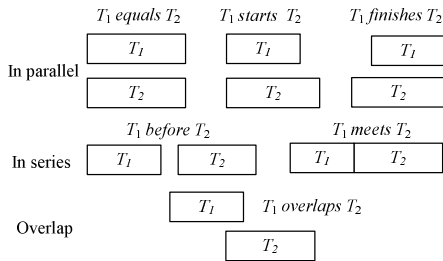


Fig. 1. Some of temporal dependencies

Iteration is a typical characteristic of PD process. In this paper, we assume that task iteration is generated due to the following causes [10]: (1) receiving new information from overlapped tasks after starting to work with preliminary inputs; (2) change of inputs, such as upstream tasks being reworked or new specifications brought forward; (3) a task's failing to meet established criteria. In our model, we assume tasks will iterate according to a certain probability.

Resource dependencies are complementary to temporal ones. They deal with the distribution of resources among tasks. Three basic resource dependencies are defined [2]: sharing, simultaneity, and volatility.

## 5 Definition of HTCPN

Although classic colored Petri nets [8] allow for a succinct description of PD processes, precise specifications for real process systems have a tendency to become large and complex. At one level we want to give a simple description of the process without having to consider all the details. At another level we want to specify the more detailed behavior of the process. This is the reason we provide a hierarchy construct, called a subnet. A subnet is an aggregate of a number of places, transitions, and sub-



systems. Such a construct can be used to structure large processes. In PD process modeling, it is very important to describe the temporal behavior of the process, i.e., we need to model durations and delays of the activities in the model.

In keeping with these ideas, this paper recommends Hierarchical Timed Colored Petri Nets (HTCPN). The HTCPN has a set of *substitution transitions*. Substitution transitions allow the embedding of a small subnet within a larger Petri net for higher level modeling. When a substitution transition needs to be fired, the underlying Petri net ought to be evaluated in order to determine the outcome of the substitution transition. When relating the individual CPN to a substitution transition, the individual non-hierarchical CPN in a HTCPN is called a *page*. The input and output places of the page are called *port nodes*. The port nodes specify the type and the number of tokens that can be introduced into the page. Each of the substitution transitions is called a *super node* and the page that contains these super nodes is called a *super page*. In a super page, the places that are connected to the super nodes (substitution transitions) are called *socket nodes*. The *socket nodes* of a super node (substitution transition) are related to the port nodes of the page through the port assignment function.

To define HTCPN, we should first introduce some syntax for writing the expressions. The *type of variable*,  $v$ , is denoted by  $\text{Type}(v)$ . The *type of expression*,  $\text{expr}$ , is denoted by  $\text{Type}(\text{expr})$ . The *set of variables in an expression*,  $\text{expr}$ , is denoted by  $\text{Var}(\text{expr})$ . A *binding of a set of variables*,  $V$ , associating with each variable  $v \in V$ , is denoted by  $b(v) \in \text{Type}(v)$ . The *value obtained by evaluating an expression*,  $\text{expr}$ , in a *binding*,  $b$ , is denoted by  $\text{expr}\langle b \rangle$ .

**Definition 1:** A HTCPN is a tuple  $\text{HTCPN} = (PG, \Sigma, P, T, A, N, C, G, E, I, R, r_0)$  where:

- (i)  $PG$  is a finite set of pages such that: (a) each page  $\text{pg} \in PG$  is a non-hierarchical CPN and, (b) none of the pages has any net element in common;
- (ii)  $\Sigma$  is a finite set of non-empty timed or untimed **types**, also called color sets.
- (iii)  $P$  is a finite set of **places**.  $P = P_o \cup P_p \cup P_s$ , where  $P_o$  is a set of ordinary places;  $P_p$  is a set of port nodes (places);  $P_s$  is a set of socket nodes (places).
- (iv)  $T$  is a finite set of **transitions**.  $T = T_1 \cup T_2 \cup T_3 \cup T_4$ , where  $T_1$  is a set of ordinary activity transitions;  $T_2$  is a set of timed transitions;  $T_3$  is a set of hierarchical transitions;  $T_4$  is a set of timed hierarchical transitions. The four types of transitions are denoted differently as showed in Figure 2.
- (v)  $A$  is a finite set of **arcs** such that:  $P \cap T = P \cap A = T \cap A = \emptyset$ .
- (vi)  $N$  is a **node** function. It is defined from  $A$  into  $P \times T \cup T \times P$ .
- (vii)  $C$  is a **color** function. It is defined from  $P$  into  $\Sigma$ .
- (viii)  $G$  is a **guard** function. It is defined from  $T$  into expressions such that:

$$\forall t \in T: [\text{Type}(G(t)) = B \wedge \text{Type}(\text{Var}(G(t))) \subseteq \Sigma], B = \{true, false\} \tag{1}$$

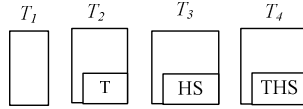
- (ix)  $E$  is an **arc expression** function. It is defined from  $A$  into timed or untimed expressions such that:

$$\forall a \in A: [\text{Type}(E(a)) = C(P)_{MS} \wedge \text{Type}(\text{Var}(E(a))) \subseteq \Sigma] \tag{2}$$

- (x)  $I$  is an **initialization** function. It is defined from  $P$  into timed or untimed closed expressions such that:

$$\forall p \in P: [\text{Type}(I(p)) = C(P)_{\text{MS}} \tag{3}$$

- (xi)  $R$  is a set of **time values**, also called **time stamps**. It is a subset of  $\mathbf{R}$  closed under  $+$  and containing 0;
- (xii)  $r_0$  is a element of  $R$ , called the **start time**.



**Fig. 2.** The four types of transitions

The set of binding  $B(t)$ , token elements TE and binding elements BE are defined in an analogous way as for CPN [8].

**Definition 2:** A step  $Y$  is **enabled** in a state  $(M_1, r_1)$  at time  $r_2$  if equation (4) is satisfied, and  $r_2$  is the smallest element of  $R$  but not less than  $r_1$ .

$$\forall p'' \in P: \sum_{\substack{(t',b) \in Y \\ p' \in p''}} E(p', t')_{r_2} < b \leq M_1(p'') \tag{4}$$

**Definition 3:** When a step  $Y$  is enabled in a state  $(M_1, r_1)$  at time  $r_2$ , the state  $(M_1, r_1)$  may change to another state  $(M_2, r_2)$ , where  $M_2$  is defined by:

$$\forall p'' \in PG: M_2(p'') = (M_1(p'') - \sum_{\substack{(t',b) \in Y \\ p' \in p''}} E(p', t') < b > r_2) + \sum_{\substack{(t',b) \in Y \\ p' \in p''}} E(t', p') < b > r_2) \tag{5}$$

We say that  $(M_2, r_2)$  is **directly reachable** from  $(M_1, r_1)$  by the occurrence of the step  $Y$  at time  $r_2$ . This is written as  $(M_1, r_1) [Y (M_2, r_2)$ .

## 6 PD Process Modeling

PD process is characterized by phase, hierarchy, iteration and prerelease, which can be described by HTC PN. Using a pump development process model, as shown in Figure 3, we can use timed hierarchical transitions to represent phase activities in PD process. In our model, each transition is marked with an HS-tag or THS-tag indicating that it is a substitution transition, and can be replaced by subnets. The timed hierarchical transition of *detail design* can be replaced by the subnet shown in Figure 4.

When transition *detail design* receives the design produced by *conceptual design*, the token can fire the transition of *TD*. The transition is a timed transition. When the time has passed, i.e., task decomposition has been finished, the transition can produce a token in the places  $P_2, P_3, P_4, P_5$  respectively. The four tokens can fire the four transitions (*PD, OVD, CD, and PBD*) simultaneously, which means that the four tasks can be performed concurrently. The four transitions are timed hierarchical transitions, and each of the four transitions can be replaced by their corresponding subnets, where

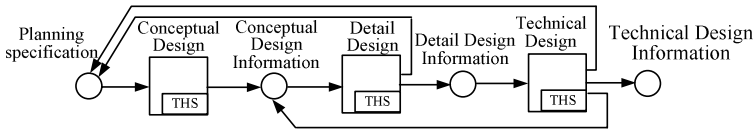


Fig. 3. Part of PD process (without arc expressions)

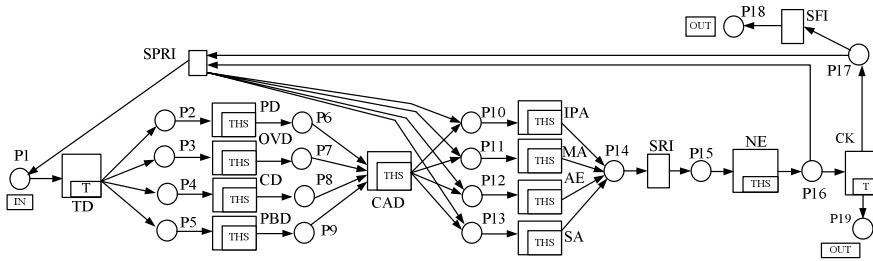


Fig. 4. Subnet of "Detail Design" (without arc expressions)

Table 1. The semantic list of notation in Fig.4

place	Semantic	transition	Semantic
P <sub>1</sub>	start detail design	TD	task decomposition
P <sub>2</sub>	start plunger design	PD	plunger design
P <sub>3</sub>	start outlet valve design	OVD	outlet valve design
P <sub>4</sub>	start cam design	CD	cam design
P <sub>5</sub>	start pump body design	PBD	pump body design
P <sub>6</sub>	plunger design information	CAD	CAD design
P <sub>7</sub>	outlet valve design information	IPA	injection performance analysis
P <sub>8</sub>	cam design information	MA	manufacturability analysis
P <sub>9</sub>	pump body design information	AE	assemblability evaluation
P <sub>10</sub>	start injection performance analysis	SA	serviceability analysis
P <sub>11</sub>	start manufacturability analysis	SRI	send revision information
P <sub>12</sub>	start assemblability evaluation	NE	negotiation
P <sub>13</sub>	start serviceability analysis	CK	check
P <sub>14</sub>	completion or revision design in-SFI formation	in-SFI	send feedback information
P <sub>15</sub>	ready for negotiation	SPRI	send part revision information
P <sub>16</sub>	negotiation information		
P <sub>17</sub>	redone information		
P <sub>18</sub>	planning & specification		
P <sub>19</sub>	detail design information		

their output places, transition *CAD* can be fired. It means that we can execute the task *CAD Design* when the four tasks have been accomplished. Just as mentioned before, it can be replaced by a subnet which describes its processes more concretely. All the other places and transitions work in a similar way. The token in  $P_{17}$  can fire the transition *SPRI* or *SFI*. Transition *SPRI* can bring iteration of its own stage, and transition *SFI* can send the feedback information to the conception design stage, which may bring iteration of conception design and detail design. Otherwise, if the place  $P_{19}$  gets a token, it means the stage detail design is executed successfully.

Our process model is based on the point of design decision-making in PD, however, details of the coordination mechanism, resource constraints and distribution among tasks are beyond this paper's scope. If we want to take into account resource constraints and distributions, we should link the transitions of tasks which require resources with the resource management system; in that way, the tasks can't be executed until the resource is available for them.

## 7 Performance Analysis of Model

The greatest advantage of our PD process modeling using HTCPN is that we can analyze both the structure of the model and its performance. This is indispensable in PD.

### 7.1 Analysis of HTCPN

An HTCPN is a specific high level Colored Petri net (CPN), so almost any analysis technique on CPN can be adapted to it. There are many CPN-based analysis techniques for verifying workflow process definitions.

Reachability graphs and place invariants [11] are two important structural analysis methods, which can be used independently of CPN to verify most dynamic properties, i.e. reachability, boundedness, liveness and fairness. That is why we can use these methods to analyze the properties of our HTCPN model.

Like other formal analysis methods, our HTCPN can be simulated by CPN software, such as CPN tools [12]. The tools can simulate the execution of actual processes to find errors in logic, which are hard to verify with the above formal methods.

All of these methods of analysis make HTCPN convenient for constructing large PD systems. As well, fewer mistakes occur in modeling when it is used.

### 7.2 Process Model Performance Analysis

In our PD process model based on HTCPN, we can easily obtain many of the results with which PD process modeling is primarily concerned:

(1) PD time analysis. PD time analysis is one of the most important aspects of the PD process, and the earliest time to market is an important goal in our modern competitive environment. In our model, each of the important transitions is a timed or timed hierarchical transition. If we estimate the period of time and possibility of failure of each transition, the overall PD time can be estimated.

(2) Critical path detection. Through the firing sequence, we can get the optimal PD executing path, and we can also discover how to achieve our expected product design process path. In order to shorten the overall PD process lead time, we should reduce the duration of tasks on critical paths. Moreover, these tasks should be given more attentions in PD process control.

(3) Deadlock analysis: Deadlock occurs if our initial marking can't achieve the final marking, or all transitions can't be fired. By simulation, the PD model based on HTCPN can help us find and/or prevent deadlock.

(4) Conflict analysis. In our PD process model, when two or more tasks produce conflict design results, there is a possibility of conflict. The graph nature of Petri nets makes it easy to detect the birth of conflict. A well established coordination mechanism is badly needed when conflict occurs.

(5) PD cost analysis. As with PD time analysis, we can relate each transition to a PD cost and can compute the cost of the entire PD. Obviously, we should estimate the cost of each design task of each stage.

(6) Resource allocation analysis. Although resource application and allocation is beyond the scope of this paper, we can use our model to analyze the performance of resource allocation and usage. The resource manager of the PD process model should assign the right resource to the right task at the right time in order to enhance design efficiency.

## 8 Conclusions

In this paper we discussed the various benefits of PD process modeling, and presented the characteristics of a modern PD pattern in order to provide an exact and full description of it. In modern PD processes, projects should be decomposed into many subtasks. Of the two types of task dependencies adopted, we discussed mainly the temporal dependencies on which product process modeling was founded. Hierarchical Timed Colored Petri Nets, a powerful modeling language, was introduced. HTCPN, derived from the classic colored Petri nets, can exactly describe the actions and information flow in PD. Ultimately, the performance analysis of HTCPN and process model based on HTCPN were presented. Various performance measures of PD process can be generated from the proposed model.

## Acknowledgments

This research was partially supported by the National Natural Science Foundation of China under the contract number 50175010, the Excellent Young Teachers Program of Ministry of education under the contract number 1766, the National Excellent Doctoral Dissertation Special Foundation under the contract number 200232, and Guangxi Liu Gong Machinery Co., Ltd, P. R. China. The authors would also like to acknowledge constructive revision suggestions and help with English by Prof. Ming J. Zuo, Department of Mechanical Engineering, University of Alberta, Edmonton, Alberta, Canada.

## References

1. Zakarian, A., Kusiak, A.: Analysis of Process Models. *IEEE Transactions on Electronics Packaging Manufacturing*. 23 (2000) 137-147
2. Alberto B. R., Adailton J. A., Christian M. A. etc.: Coordination Components for Collaborative Virtual Environments. *Computers & Graphics*. 25 (2001) 1025-1039
3. Liu D., Wang J., Stephen C.F. etc.: Modeling Workflow Processes with Colored Petri Nets. *Computers in Industry*. 49 (2002) 267-281
4. Zhao L., Jin Y.: Modeling Collaborative Engineering Design Process Using Petri-Net. *Proceedings of DETC'00: 2000 ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Baltimore, Maryland (2000)
5. Jensen K.: *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*. Vol. 3, Practical Use. *Monographs in Theoretical Computer Science*. 2nd edn. Springer-Verlag, Berlin Heidelberg New York (1997)
6. Jiang Z., Zuo M.J., Fung R.Y.K. etc.: Colored Petri Nets with Changeable Structures (CPN-CS) and their Applications in Modeling One-of-a-Kind Production (OKP) Systems. *Computers and Industrial Engineering*. 41 (2001) 279-308
7. Jiang Z., Zuo M.J., Fung R.Y.K. etc.: Temporized colored Petri Nets with changeable structure (TCPN-CS) for performance modeling of dynamic production systems. *International Journal of Production Research*. 38 (2000) 1917-1945
8. Jensen K.: *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*. Vol. 1, Basic Concepts. *Monographs in Theoretical Computer Science*. 2nd edn. Springer-Verlag, Berlin Heidelberg New York (1997)
9. Negele H., Fricke E., Schrepfer L. etc.: Modeling of Integrated Product Development Processes. *Proceedings of the 9th Annual Symposium of INCOSE, UK* (1999)
10. Cho S.H., Eppinger S.D.: Product Development Process Modeling Using Advanced Simulation. *Proceedings of DETC'01 ASME 2001 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Pittsburgh, Pennsylvania (2001)
11. Jensen K.: *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*. Vol. 2, Analysis Methods. *Monographs in Theoretical Computer Science*. 2nd edn. Springer-Verlag, Berlin Heidelberg New York (1997)
12. CPN group at the Department of Computer Science, University of Aarhus, Denmark. Online at: <http://www.daimi.au.dk/CPnets/>, (2005)

# An Intelligent Petri Nets Model Based on Competitive Neural Network

Xiao-Qiang Wu

School of Economics & Management, Beijing University of Aeronautics & Astronautics,  
100083 Beijing, PR China  
wuxiaoqiang@buaa.edu.cn

**Abstract.** Petri nets are powerful and versatile tools for modeling, simulating, analyzing, designing and controlling of many complex systems. This paper addresses a hybrid approach combining competitive neural network and Petri nets in the formal model of intelligent Petri nets. The proposed model not only takes the descriptive advantages of Petri nets, but also has neuron learning and knowledge reduction ability like competitive neural network. It is suitable for dynamic process and information, e.g., the weights are adjustable. For the simulation of the model, it is applied in the conceptual modeling of supply chain for inter-organizational cooperation in manufacturing industry. The intelligent Petri nets model is an innovative method concerning intelligent transition of Petri nets. Meanwhile, the numerical example illustrates that the proposed model can be applied to the real-time supplier selection for intelligent decision-making, and a novel method is provided for modeling in supply chain concerned.

## 1 Introduction

Petri nets have been widely applied in modeling and controlling discrete event distributed systems [1-3]. Petri Nets (PNs) have the ability to represent and analyze in an easy way concurrency and synchronization phenomena. However, as for general dynamic systems, PNs have some disadvantages in this application field. PNs cannot represent data flow, and when data logic and control logic are different, it cannot apparently represent data flow that is independent of control flow. On the other hand, neural network is well known to have advanced learning and representation ability [2-8]. Additionally, Competitive Neural Network (CNN) has a dynamic learning ability.

Furthermore, PN approach can be easily combined with other techniques or theories such as object-oriented programming, neural networks and others. PNs have an inherent quality in representing logic in an intuitive and visual way. An adaptive fuzzy Petri net has been presented and used for knowledge representation and reasoning in [2] under a more generalized reasoning rule. Kotaro proposed a Learning Petri Network (LPN) for application into nonlinear control systems [6]. Donald put forward neural Petri nets and modeled the control of track switch of the railroad [7]. Jaya and Amit [9] discussed an algorithm of machine learning for the neural Petri net by training a multilayered feed-forward neural net with membership distribution of the input and the output variables as the training instances.

Accordingly, it is very important to design a dynamic and intelligent Extended Petri Nets approach that is adjustable according to process variation of general dynamic systems. Aiming at this object, Extended Petri Nets (EPN) model based on Competitive Neural network (CNN) is proposed in this paper. It is called Intelligent Neural Extended Petri Nets (INEPN).

## 2 Preliminaries

### 2.1 Definition of Extended Petri Nets

The modeling methods of Petri nets are based on states, and PNs have more abundant expressions and more flexible properties. The only easy way for the implementation of inner operations and responses to exception is to treat with tokens and fires in the network, which makes it suitable for the modeling of inter-organizational systems.

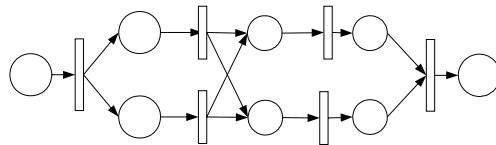


Fig. 1. An example of extended Petri nets

The traditional way to express the complex processes in an enterprise through PNs makes the model excessively complicated. Therefore, we extend the traditional PNs approach in following three ways. An example of extended Petri nets can be shown in Figure 1.

(1) Colored Petri nets extension. Colored Petri nets model can represent classes of business processes. Therefore, the states of places are the existence or nonexistence and the weight on arc is 1. Each appearance of a token represents an instance and it must have an identifier. When transition happens, prediction can be made to determine which token is subjected. The different colored tokens in initializing condition mean they need to deal with different instances.

$$EPN=(P,T,F,color) \tag{1}$$

where,  $P=\{p_1,p_2,\dots,p_n\}$  is a finite set of places,  $n \geq 0$  ;

$T=\{t_1,t_2,\dots,t_m\}$  is a finite set of transitions,  $m \geq 0$  ,  $P \cup T \neq \emptyset$  and  $P \cap T = \emptyset$  ;

$F=\{f_1,f_2,\dots,f_j\}$  is a finite set of flows,  $j \geq 0$  ,  $F \subseteq (P \times T) \cup (T \times P)$  ;  $color$  is a set of the colored instances.

(2) Time factor extension to represent transition execution time or transition delay.

$$EPN=(P,T,F,color,PS,TS) \tag{2}$$



where  $PS$  is execution time transition consumes on places;  $TS$  is the delay time of transition.

(3) Object Oriented (OO) extension through many subsidiary nets, which can present the workflow in each organization separately, and each subsidiary net can communicate synchronously or asynchronously and work cooperatively.

$$EPN=(EPN_1,EPN_2,\dots,EPN_n) \tag{3}$$

where  $n$  is the number of Petri net instances involved.

### 2.2 Fundamentals of CNN

The learning algorithm for competitive Neural Network simulates the neurons system of biology, which depends on the dynamic mechanism that relies on the simulation, cooperation and inhibition, competition to process information, to direct the learning and work of the nets. And it does not like the most of the neural networks, which take the errors of network or the function of energy as the rules of the algorithm. Competitive neural network can construct some kinds of nets which have the ability of self-organizing.

There are many forms and algorithms for competitive neural networks. However, the basic construction is mostly used. There are neurons in the input layer and neurons in the output layer. The synaptic weight of the network is  $w_{ij}(i=1\sim n, j=1\sim m)$ .

The constraint condition is:

$$\sum_{i=1}^n w_{ij}=1 \tag{4}$$

Input

$$U_k=[u_1^k \quad u_2^k \quad \dots \quad u_n^k]^T \tag{5}$$

Output

(6)

$$V_k=[v_1^k \quad v_2^k \quad \dots \quad v_m^k]^T, (k=1\sim p) \tag{6}$$

## 3 The CNN-Based Intelligent Petri Nets Model

**Definition 1:** In INEPN, all the input and output places can be defined as:

$$P_{in}=\{p \in P | \exists (p, t) \in Z\}$$

$$P_{out}=\{p \in P | \exists (t, p) \in Z\}$$

All the input and output transitions can be defined as:

$$T_{in} = \{t \in T \mid \exists (t, p) \in Z\}$$

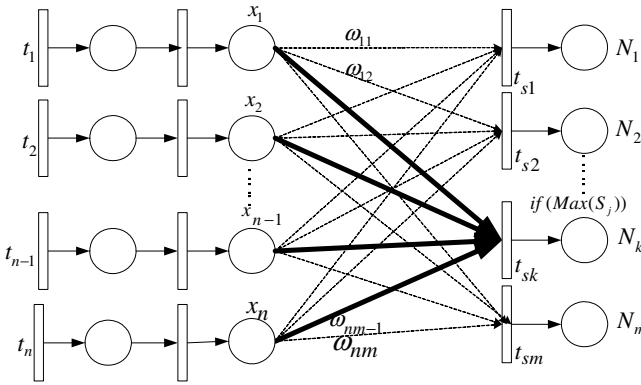
$$T_{out} = \{t \in T \mid \exists (p, t) \in Z\}$$

**Definition 2:**  $\forall t \in T, \forall p_{ij} \in T_{in}, j = (1, \dots, n)$ , if the given  $t$  is activated, the input  $S_j$  of all the neurons respectively in the competitive layers can be addressed as:

$$S_j = \sum_{i=1}^n w_{ij} u_i^k, (j=1, 2, \dots, m) \tag{7}$$

**Definition 3:** In the rule WTA (Winner Takes All), the neuron whose corresponding input  $S_j$  is maximum is the winner, therefore its output value is put into 1, otherwise the others' output value are put into 0, shown in Figure 2, i.e.,

$$\begin{cases} v_j = 1 & S_j > S_i \quad (i=1, 2, \dots, m; i \neq j) \\ v_i = 0 & (i \neq j) \end{cases} \tag{8}$$



**Fig. 2.** An INEPN based on CNN

INEPN is a hybrid model of competitive neural network and extended Petri nets. In mathematical terms, INPEN model can be described as:

$$INEPN = (EPN_1, EPN_2, \dots, EPN_n, INEPN_1, INEPN_2, \dots, INEPN_m) \tag{9}$$

where  $EPN$  as described in formula (2)

$$INEPN_i = (P, T, F, color, PS, TS, U_k, V_k, W) \tag{10}$$

where  $n$  : the number of subsidiary Petri nets instances involved,  $m$  : the number of subsidiary INEPN instances involved  $U_k, V_k$ , see formula (5) and (6) respectively. Here,  $P = \{P_{in}, P_{out}\}$ , and  $T = \{T_{in}, T_{out}\}$ . In formula (10), a matrix  $W$  is called a place to transition connectivity matrix,  $\omega_{ij}$  is synaptic weight:

$$W = \begin{Bmatrix} \omega_{11} & \omega_{12} & \cdots & \omega_{1m} \\ \omega_{21} & \omega_{22} & \cdots & \omega_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_{n1} & \omega_{n2} & \cdots & \omega_{nm} \end{Bmatrix},$$

At the same time  $\omega_{ij}$  must satisfy with formula (4).

### 4 The Learning Process and Solution of the Model

In INEPN, the following learning algorithm should be kept:

**Step 1.** Initialization. Stochastic value should be given to  $w_{ij}$  between [0,1] according to the constraint condition (see formula (4)).

**Step 2.** Select a pattern  $U_k$  from  $p$  input patterns as the input layer of neural network.

**Step 3.** Compute the input  $S_j$  of all the neurons respectively in the competitive layers according to Definition 2.

**Step 4.** Compute  $v_j$  according to Definition 3, and transition  $t$  is deduced.

**Step 5.** Updates the winner neuron’s synaptic weight according to following formulas otherwise keeps no changes.

$$w_{ij} = w_{ij} + \Delta w_{ij} \tag{11}$$

$$\Delta w_{ij} = \eta \left( \frac{u_i^k}{M} - w_{ij} \right) \tag{12}$$

where  $i=1,2,\dots,n$ ,  $\eta$  is learning rate parameter ( $0 < \eta < 1$ ), generally takes  $\eta$  from  $0.01 \sim 0.03$ .  $M$  is the number of elements that are equal to 1 in learning pattern vector

$$U_k = [u_1^k, u_2^k, \dots, u_n^k]^T.$$

**Step 6.** Choose another learning pattern and return to Step 3 until all of learning patterns have been provided to net.

**Step 7.** Return to Step 2 until all the adjustment values of synaptic weights are sufficiently small.

Formula (9) is object-oriented extended INEPN model, and formula (8) is an INEPN model. The result of formula (8) is formula (10).

Pseudo code of INEPN can be described as:

```

INITIALIZE all  $\omega_{ij}$  randomly from [0,1]
WHILE  $\omega_{ij} < \theta$ 
  REPEAT
    For  $i=1,\dots,n$ 
      FOR  $j=1,\dots,m$ 

```

```

calculate weights  $\omega_{ij}$ 
calculate  $S_j$  according to Formula (7)
FOR  $j=1,\dots,m$ 
sum over index  $u$  to approximate  $\sum \omega_{ij} u_i^k$ 
FOR  $j=1,\dots,m$ 
update according to the rule Formula (12)
UNTIL converged
 $T-T \cdot \eta, 0 < \eta < 1$ 
END

```

### 5 Applications in Conceptual Modeling of Supply Chain

In this paper, we take an aluminum corporation as an example to illustrate how to apply INEPN to construct a conceptual model of supply chain. Figure 3 illustrates the sketch map of all processes including customer ordering, production process and customer acceptance.

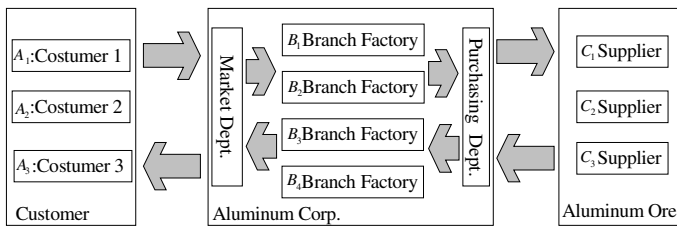


Fig. 3. Processes of aluminum corp

A conceptual model in supply chain can be constructed based on INEPN. In this model, CNN intellectualizes the real-time selection of branch factory.

Supply chain for a manufacturing enterprise can be regarded as one assembly of several different participants in different processes including ordering, product design, manufacturing, business and so on. The complexity of the production and business determines that the selection of branch factory or supplier is a kind of systemic and real-time decision-making. Inter-organizational enterprises processes have some characteristics, e.g., combination with discrete production processes, extension to the scale of management, expansion of business more agilely and flexibly, and response to market demands quickly and intelligently. The analysis of branch factory selection for manufacturing corporations is presented in Figure 4.

One of the methods dealing with selection of branch factory concerning decision-making techniques is virtual enterprise. As a result of the highly efficient cooperation among organization members, the virtual enterprise completes tasks by teaming up with different organizations and business processes successfully. Under a virtual enterprise environment, various cooperators with different benefits are highly

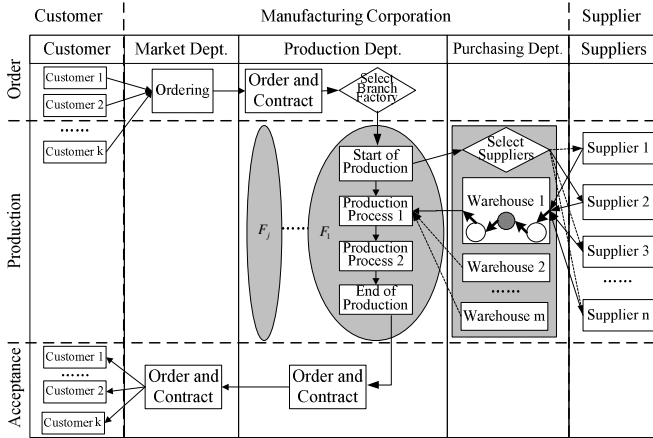


Fig. 4. The analysis of branch factory selection for manufacturing corporations

autonomous and the relations will be equal among organizations or between subsidiary companies and parent companies after the agreement is reached.

**Numerical Example:**

For the convenience of analysis, we suppose that market department is responsible for gathering the customers’ order forms and transferring demand plans to branch production factory to manufacture, whereas purchasing department’s responsibilities are supplying the raw materials (e.g. fuels and aluminum ores, and others.) Air-conditioner manufacturers, planemakers and aluminum refinement corporations are customers of aluminum corporation, distributed in region  $A_1$ ,  $A_2$  and  $A_3$  separately.  $B_1$ ,  $B_2$ ,  $B_3$  and  $B_4$  are branch factories scattered in region  $B_1$ ,  $B_2$ ,  $B_3$  and  $B_4$  correspondingly.  $C_1$ ,  $C_2$  and  $C_3$  are the material suppliers.

Figure 4 shows processes of aluminum corporations. The intelligent selection of branch factory model in a manufacturing supply chain based on INEPN, is demonstrated as Figure 5. In this example, we apply CNN in intelligentizing the transition condition of the process to select branch factory.

Generally, the following factors can affect the selection of branch factory: production capability, raw material price, transport costs and convenience, productive rate, delivery time, seasonal factors, order quantity, productive cycle, climates and others. In this model, these factors are quantified and taken as input  $U_k$ . Let  $\eta=0.15$ , and  $\omega_{ij}$  can be drawn through the learning of CNN. The calculating results can be expressed as table 1 and Figure 6. From these calculating results, we can see that the INEPN learning algorithm is effectively if we can suppose the initialized weights are appropriate.

The proposed INEPN model is performance-wise better than the traditional back-propagation algorithm. Time-complexity of the INEPN model is less than the BP algorithm. The absolute convergence of the INEPN model occurs after adaptation of

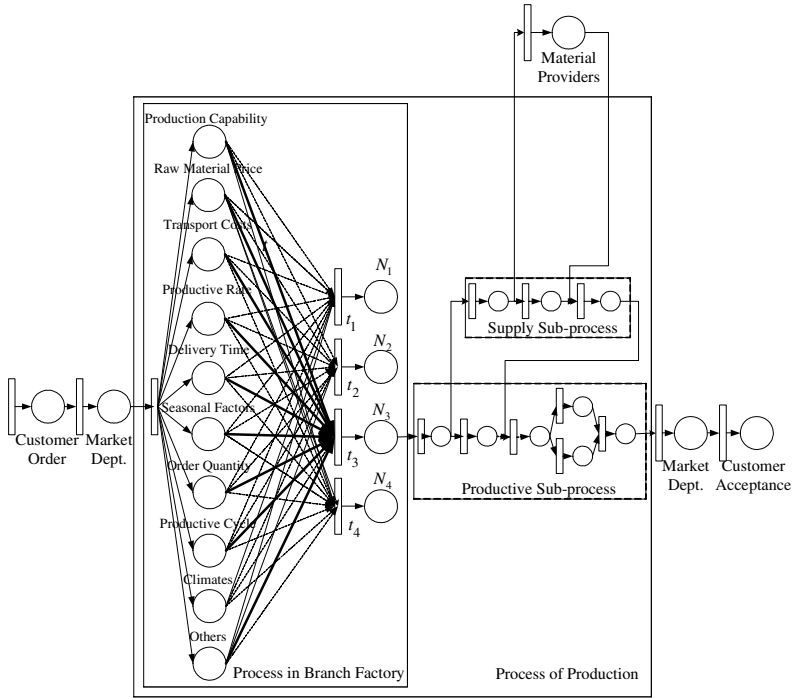


Fig. 5. An intelligent branch factory selection model applying INEPN

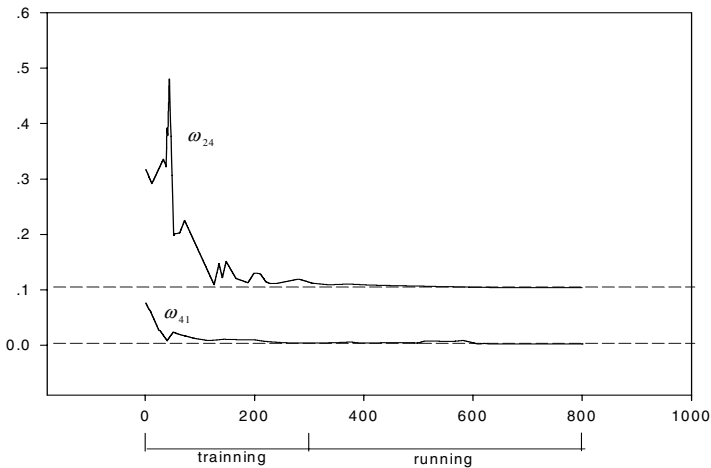


Fig. 6. Progress of weights learning in INEPN model

the network thresholds less times at all layers than BP algorithm and once later in the output layer. Such absolute convergence guaranteed even for a relative rough error surface.

**Table 1.** The calculating results of  $\omega_{ij}$ 

$\omega_{ij}$ \ i \ j	1	2	3	4	5	6
1	0.1030	0	0.0322	0	0	0.1200
2	0.1127	0.0121	0	0.1042	0.2012	0.0238
3	0.1018	0	0.0122	0	0.1003	0
4	0.0023	0.1121	0.0017	0.1134	0.0019	0.0897
5	0.1015	0.1020	0.1126	0.1147	0.0002	0.0023
6	0.0213	0.1023	0.1034	0.0098	0.0789	0
7	0.0224	0.0341	0.1134	0.0128	0.1023	0
8	0.1013	0.0020	0.0004	0	0	0.1101
9	0.0097	0.1009	0.0056	0.1121	0.1118	0.1016
10	0.0006	0.0082	0.0927	0	0	0.1023

## 6 Conclusions

This paper presents a CNN-based intelligent Petri nets model and its application in supply chain conceptual modeling. A numerical example is used for illuminating the calculating and learning process of the proposed model. It is a supplement for intelligent Petri nets research. This model gives traditional Petri nets the ability of learning and intelligent decision-making. The idea proposed in this paper is a new formal way to solve the Petri nets learning problem.

In addition, it can also be taken as a new modeling method for supply chain theory and practice. Many researches are carried out on how to define the upstream and downstream members of the supply chain [10]. What should be emphasized is that the proposed model can be applied to the real-time supplier selection for intelligent decision-making.

## References

1. Hoffner, Y. Lwdwig, H. Gulcu, C. Grefen, P.: An Architecture for Cross-Organizational Business Processes. Proc. Second International Workshop on Advance Issues of E-Commerce and Web-Based Information Systems, (2000) 2-11.
2. Xiaou, L., Wen, Y., Felipe, L.-R.: Dynamic Knowledge Inference and Learning under Adaptive Fuzzy Petri Net Framework. IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and reviews 4 (2000) 1206-1212
3. Kevin, F.R., Liu, C.H., Chung, L.: High-Level Fuzzy Petri Nets As a Basis for Managing Symbolic and Numerical Information. Int. J. on Artificial Intelligence Tools, 4 (2000) 569-588
4. Omer, F.R.: Automating Parallel Implementation of Neural of Neural Learning Algorithms. Int. J. of Neural Systems, 3 (2000) 227-241
5. Jang, J.S.R., Sun, C.T.: Neuro-fuzzy modeling and control. Proc. IEEE 3 (1995) 378-405
6. Kotaro, H., Masanao, O., Singo, S., Hu, J.L.: Learning Petri Network and Its Application to Nonlinear System Control. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics 6 (1998) 781-789
7. Donald, T., Irvin, R.J.J.: Synthesis of Intelligent Switching Systems Using Neural Petri Nets. Proc. IEEE Int. Conf. on Systems, Man and Cybernetics, 2003. (2003) 1298-1303

8. Hanna, M.: Modeling Product Quality in a Machining Center Using Fuzzy Petri Nets with Neural Networks. Proc. IEEE Int. Conf. on Robotics & Automation (1999) 1502-1507
9. Jaya, S., Amit, K.: A Hybrid Approach to Knowledge Acquisition Using Neural Petri Nets and DS Theory. Int. J. of Computational Intelligence and Applications 2 (2001) 203-223
10. Hokey, M., Zhou, G.G.: Supply Chain Modeling: Past, Present and Future. Computers & Industrial Engineering (2002) 231-249



# An Automatic Coverage Analysis for SystemC Using UML and Aspect-Oriented Technology

Yan Chen<sup>1</sup>, Xuan Du<sup>2</sup>, Xuegong Zhou<sup>1</sup>, and Chenglian Peng<sup>1</sup>

<sup>1</sup> Department of Computing and Information Technology, Fudan University,  
Shanghai 200433, China

{021021099, 022021130, c1peng}@fudan.edu.cn

<sup>2</sup> Network Enterprise Department, Zhongxing Telecommunication Equipment Corporation,  
Shanghai 201203, China  
du.xuan@zte.com.cn

**Abstract.** SystemC can be considered as the best possible language today for system level design and exploration of embedded systems. However, testing SystemC descriptions is still an open issue, since the language is new and researchers are looking for efficient error models and coverage metrics, which can be indifferently applied to hardware and software modules. In this paper we propose a novel approach to automate the test coverage analysis for SystemC descriptions using UML and Aspect-Oriented technology. SystemC meta-model and aspect meta-model are established to support UML customization and extension. They also provide the foundation for aspect weaver and SystemC code generator. Expected functional coverage metric could be extracted from UML timing descriptions so that it is possible to automate the whole test coverage analysis. By using the aspect-oriented technology test functionalities could be added or replaced without modifying the original design. It makes system designs more readable and easier to maintain.

## 1 Introduction

Today's embedded systems consist of multiple architectural components including software and hardware. The ability to specify and verify these systems at a high level of abstraction is a key competence to cope with the increasing design complexity. C/C++-based approaches on system specification and design are becoming more and more important. The leading approach for C++-based system specification is SystemC [1], which is on the step of becoming a *de facto* standard in industrial system-level design. SystemC provides efficient and accurate models of hardware/software components and allows a high performance simulation of system behavior during the whole design process.

Considering that functional verification has become a real bottleneck of the entire design process, accurate verification methodologies are needed. However, testing SystemC descriptions is still an open issue, since the language is new and researchers are looking for efficient error models and coverage metrics, which can be indifferently applied to hardware and software modules. SystemC models are

executable descriptions so that they can be easily integrated with simulation-based verification. In current industrial practice, simulation-based verification consists of the generation and simulation of massive amounts of random tests. Advanced random generators can improve the quality of generated tests, but cannot detect areas in the design that are not tested while others are tested repeatedly. Coverage analysis with some well-defined coverage metrics performs a quantitative analysis of simulation completeness [2][3]. With the coverage reports, verification resources can be steered toward areas of low coverage, making verification efforts more effective. Now the main approach for coverage analysis is to create a comprehensive list of tasks which are inserted into the source code and then check that each task is to be covered during verification. However, using this method designers are always be troubled with the interweave of both design code and test code so that it is difficult to reuse and change the original design code.

This paper presents a novel coverage analysis method which applies aspect-oriented programming (AOP) [4] to SystemC descriptions and helps out of the above problem. Aspect-oriented programming (AOP) is a new programming technique that supports to encapsulate *aspect code* which is formerly tangled with the normal *component code* and can weave such crosscutting concerns into original code automatically with the help of an aspect weaver. Verification procedure can benefit greatly from AOP because constraint checkers and debugging codes can be separated and modularized from the main functionality, and in the meanwhile it is much more flexible to meet various requirements of testers.

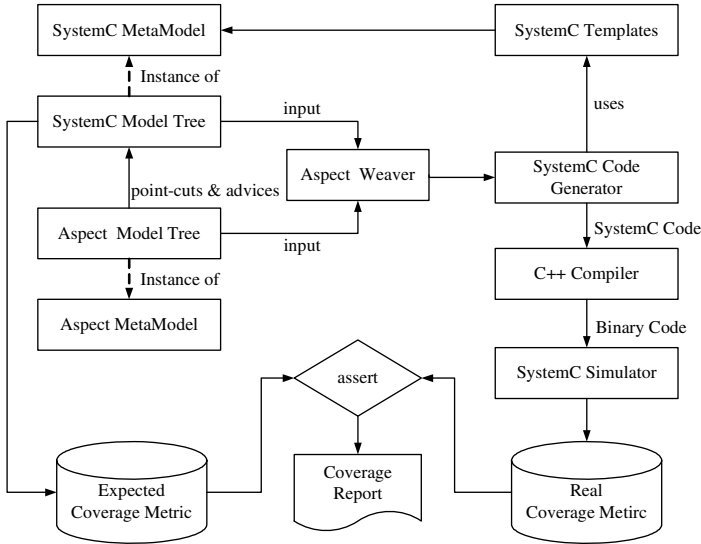
The two important components of AOP are *pointcuts* and *advices*. Pointcuts correspond to points in the dynamic execution of the program. They can often be formulated in terms like: before or after a method (constructor, process) is called; if any of exceptions takes place; when something (attribute or variable) gets changed, etc. Once a *pointcut* is reached an *advice* will be triggered. Advices execute relative actions that one wants to apply to the source code. The combination of the pointcut and the advice is termed as *an aspect*. In our paper we build the coverage checker aspects which include where (and in what cases) we want checkers to be placed and what action we want to take in these checkers.

The Unified Modeling Language (UML) [5] is currently being used as the universal technique for modeling object-oriented applications across development life cycles from design to verification. As a compromise between the requirements for a standard notation and for domain-specific modeling languages, UML was designed as an extensible modeling language with extension mechanisms. To apply UML to model SystemC and aspect, we develop a SystemC-specific meta-model and an aspect-specific meta-model to serve as formal definitions of such extensions. They add more semantic depth to the standard meta-model and thus build a foundation for model analysis, aspect weaving and code generation.

The remainder of this paper is organized as follows: In section 2, we briefly introduce our automation methodology for functional coverage analysis. Section 3 introduces UML extensions for SystemC and aspect modeling. Section 4 discusses the expected functional coverage metric. Finally, in section 5, we present a conclusion and suggest possible areas of future investigation.

## 2 Automatic Functional Coverage Analysis

In this section, we propose a method to automatically quantify functional coverage. Figure 1 illustrates our approach to automate coverage analysis.



**Fig. 1.** Functional coverage analysis automation

Recently, many adaptations and extensions to UML have been made to reflect a domain’s world view. As a technique, domain-specific UML meta-modeling has gained in importance. In our approach, SystemC meta-model and aspect meta-model are established for SystemC and aspect modeling. SystemC models describe systems while aspects models represent pointcuts and advices to the original models for verification purpose. A model is an instance of a meta-model, meaning that every element of the model is an instance of an element in the meta-model. By this formalization a model is represented as a syntax tree so that it is possible to analyze and transform the model accurately. Through an aspect weaver and a code generator, SystemC code coupled with test code are produced automatically. Then executable codes are simulated for the final coverage report.

A challenge with this approach to automate the evaluation of verification coverage is the requirement to determine which functional coverage metric to be applied on SystemC specification. Functional coverage, which as the name implies, focuses on the functionality of the design is to prove that all functions undergo simulation. Functional coverage has been proved more effective in finding errors because it focuses on application domain and can be tuned to areas which users think are of significance [6][7]. However, functional coverage is specific to each design and thus more difficult to define and measure than regular code-based coverage.

In our approach, state diagrams and sequence diagrams of the UML are used to illustrate the dynamic view of the system. Functional coverage metric essentially defines the set of paths with a finite list of points to be covered. So it is natural to concentrate on sequence diagrams to get specific coverage metrics which consist of interactions of objects and actors representing a path of the system behavior. Specifically, in UML2.0 [8], timing diagrams are used to bridge a certain gap between the sequence and state diagrams. Timing diagrams can show both of the present state of the system and the time dependencies between. These expression abilities are very useful for our coverage metric definition. More details will be discussed in section 4. It is possible to extract the expected coverage metric automatically from model trees.

In the verification procedure, the system model does not need to know about any functionality the aspect has added. When the verification is finished, the original system models bypass the aspect weaver and regain the original functionality. So using this approach verification tasks gain the most flexibility.

### 3 UML Extensions for SystemC and Aspect Modeling

The UML specification is defined by using a meta-modeling approach that adapts formal specification techniques to increase the precision and correctness of the specification. The UML meta-model conforms to a 4-layer meta-model architectural pattern [5], which are user object layer(M0), user model layer(M1), meta-model layer(M2) and meta-metamodel layer(M3) respectively. The primary responsibility of the meta-model layer is to define a language for specifying models while the meta-metamodel layer is to define the language for specifying a meta-model. The OMG has encouraged and adapted the Meta Object Facility (MOF) [9] as the standard meta-metamodel. In our paper, language extensions to UML refer to SystemC meta-model and aspect meta-model which all conform to the MOF standard.

The core language elements of SystemC include modules and ports for representing structural information, channels and interfaces as abstraction for communication, and processes for expressing concurrency behaviors, and event as a flexible, low-level synchronization primitive [1]. A complex system consists of nested modules. A meta-model formally defines the constitution and the abstract syntax of the modeling language. So we need introduce the new SystemC concepts into UML. Fig. 2 shows the meta-model for SystemC modeling. UML extension based MOF can make use of the object-oriented modeling facility, including creating new meta-classes and creating new associations between meta-classes. Existing Basic meta-classes are imported from the Core package defined in the UML2.0 [8].

We illustrate a SystemC model with a simple system. As shown in Fig. 4, there is a simplified graphic view of this system in the right and a tree data structure in the left. This system is a typical system-level description which includes a master(m1), a slave(m2) and a bus(channel). The master(m1) gets a access to interface i through port p1 while the channel gets a access to interface i2 through port p. It should be noted that every node of the tree structure is an instance of an element of SystemC meta-model.

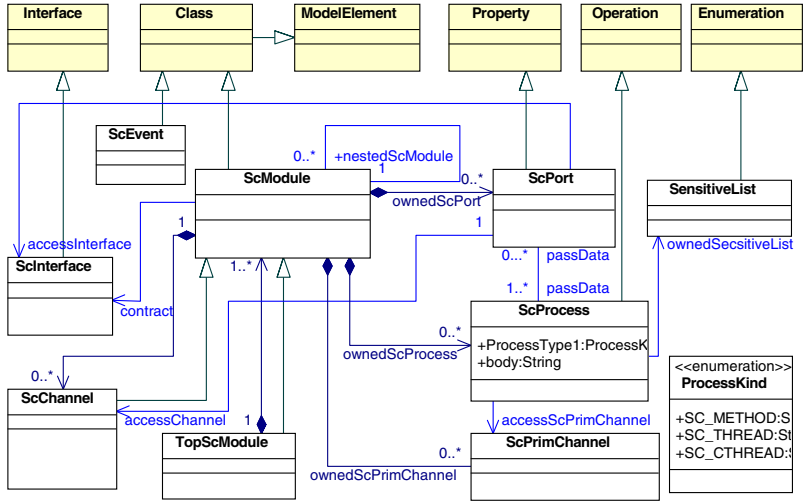


Fig. 2. Meta-model for SystemC modeling

Fig. 3 shows a meta-model for AOP modeling. The core concept of the AOP extension is modeled by the Aspect metaclass. An aspect also owns pointcuts and advices, modeled by the Pointcut and Advice metaclasses and the pointcut-aspect and pointcut-advice associations. A pointcut specifies a pattern of message interception, specified with the pointcutExpression. Finally, each advice uses exactly one pointcut, through the advice-pointcut association, and specifies some kind of action to be performed when the pointcut condition occurs, using the body attribute.

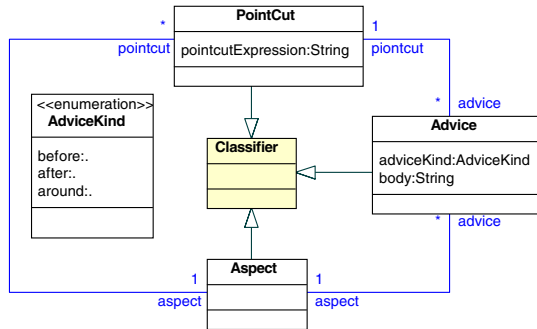


Fig. 3. Simplified meta-model of AOP modeling

In our paper, aspects refer to checkers aspects. A simple example is given and also shown in Fig. 4. This aspect intends to monitor when the wait for events statement is met and to add suitable writelog functionality. The after or before adviceKind means writelog will be executed after or before the pointcut occurs.

AspectC++ [10] is an open source aspect-oriented extension to the C++ language, so it could be an available aspect weaver at the SystemC code level. However we need an aspect weaver working at model level. We provide an aspect weaver that aims to analyze model trees and produce the results which have been added the desired verification tasks. There are two kinds of pointcuts to be dealt with by our aspect weaver. One is the pointcut to be weaved into the existed model node, for example, the pointcut when the interface method is called. The other needs to be weaved into the internal body of the model node. For example, the wait for events statement will occur within the body of the process method, so the aspect weaver adds a reference to the body of the process method. As illustrated in Fig. 4, the model tree will be transformed to the result with the shaded part after aspect weaving. AOP allows us to layer rather than embed functionality so that system models are more readable and easier to maintain.

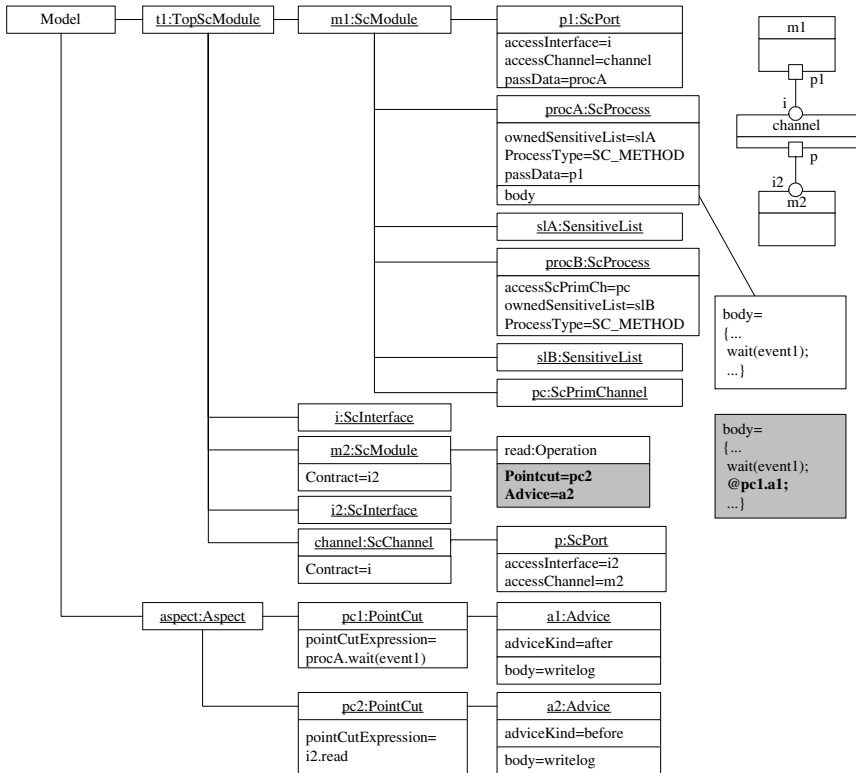


Fig. 4. An example of system model tree

Our modeling and weaving tools are implemented based on the Eclipse Platform [11]. Eclipse provides a modeling framework EMF which is an implementation of the MOF specification and can provide a highly efficient Java implementation of a core subset of the MOF API.

### 4 Expected Functional Coverage Metric

In SystemC the basic unit of functionality is called a process. A process must be contained in a module—it is defined as a member function of the module class and declared to be a SystemC process in the module’s constructor. SystemC processes are different from the regular member function of UML class because there is no explicit call on them at all. SystemC processes are triggered or resumed by the events which may occur during the course of simulation. Because a process is always event-driven or reactive, system behaviors are able to be captured by UML state or sequence diagrams.

It is possible to extract all state transition automatically from UML state diagrams but we think all path enumerations make no sense to specific system functionality. So we use timing diagram of UML2.0 to represent state transition sequence related to specific system functionality. A timing diagram shows the transition in state of an object over time in response to events or stimuli. Figure 5(a) gives a timing diagram with one module. In fact a system will be complicated since there are communication and interaction between modules. Figure 5(b) gives another timing diagram with more than one module and messages between them. Clock is a special object which will send clock-event to other modules. Clock-event will be useful when the change of variables in the static sensitivity list needs to be captured. From timing diagrams it is apparent that the communication between modules results in an expected functional coverage metric. For example, an expected state transition sequence is: (m1,sta1)-(m1,sta2)-(m2,sta1)-(m2,sta2)-(m1,sta3)-(m2,sta2)-(m1,sta1). So we can extract our expected functional coverage metric from such timing descriptions. We can designate the name of every coverage metric as the same as the use case name because every metric we discuss here responds to one certain functionality of the system.

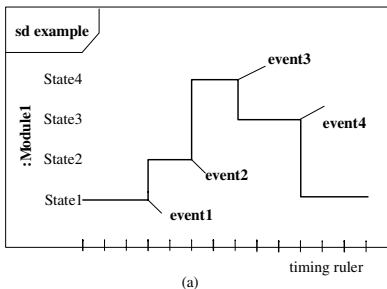


Fig. 5(a). Timing diagram with one module

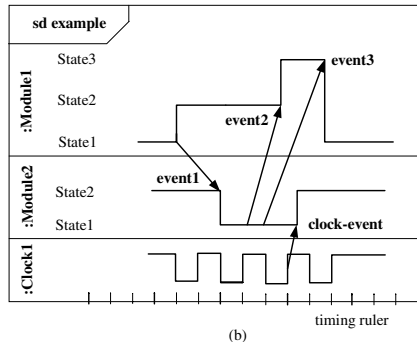


Fig. 5(b). Timing diagram with more than one module

### 5 Conclusion and Future Work

Since Aspect-Oriented Programming (AOP) is a new paradigm that shall not compete but enhance the Object-Oriented Programming (OOP), recently it has obtained more

and more attention. In this paper we mainly focus on applying AOP to the functional coverage analysis of SystemC description. SystemC can be considered as the best possible language today for system level design and exploration of embedded systems. New efficient coverage analysis method addressing with SystemC is needed.

We believe that UML modeling is essential because it is a standard modeling language across development life cycles from design to verification. In the future we plan to study SystemC code generation by further considering a whole hardware/software integrated system.

Aspect-Oriented Programming gives us a way to separate and encapsulate verification functions from the design under test (DUT). We think AOP will be encouraging not only with coverage analysis but also with other functions such as debugging and performance measurement. A further investigation would be to study how the verification aspects could be grouped into a reusable framework.

## References

1. Grotker, T., Liao, S., Martin, G. and Swan, S.: System Design with SystemC. Kluwer Publishers, (2002)
2. Gupta, S., Ashar P.: Toward formalizing a Validation Methodology Using Simulation Coverage. Proc. 34th Design Automation Conf., (1997)
3. Benjamin, M., Geist, D., Hartman, A., Wolfsthal Y., Mas G. and Smeets R.: A Study in Coverage-Driven Test Generation. Proc.36th Design Automation Conf., ACM Press, New York, (1999) 970-975
4. Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C.V., Loingtier, J., and Irwi, J.: Aspect-Oriented Programming. Proceeding of ECOOP, (1997)
5. Object Management Group: UML 2.0 Infrastructure Final Adopted Specification. <http://www.omg.org/docs/ptc/03-09-15.pdf>, (2003)
6. Grinwald, R. et al.: User Defined Coverage - A Tool Supported Methodology for Design Verification. Proc. 35th Design Automation Conf., New York, (1998) 158-163
7. Tasiran, S., Keutzer, K.: Coverage metrics for functional validation of hardware designs. IEEE Design & Test of Computers, 18(4) (2001) 36-45.
8. Object Management Group: UML 2.0 Superstructure Final Adopted Specification. <http://www.omg.org/docs/ptc/03-08-02.pdf>, (2003)
9. Object Management Group: Meta Object Facility Specification. <http://www.omg.org/docs/ptc/03-10-04.pdf>, (2003)
10. AspectC++. version 0.9, <http://www.aspcetc.org>
11. Eclipse Project. <http://www.eclipse.org>



# Optimistic Locking Concurrency Control Scheme for Collaborative Editing System Based on Relative Position

Qirong Mao, Yongzhao Zhan, and Jinfeng Wang

Dept. of Computer Science & Communication Engineering, Jiangsu University,  
Zhenjiang, 212013, P.R.China  
{mao\_qr, yzzhan}@ujs.edu.cn, jinfeng\_wang@msn.com

**Abstract.** In order to meet the requirements of high responsiveness and unconstrained collaboration in real-time collaborative editing systems, this paper proposes a novel multi-granularity optimistic locking scheme for concurrency control in collaborative editing systems based on relative position. In the proposed scheme, reading lock and editing lock are taken into account, and the start position of locking region and that of operation are relative, and they are not transformed into absolute positions until operations are sent to collaborative sites or locks are added into lock table. Additionally, the granularity of lock can be selected by co-editors optionally, and any co-editor can edit the locking region without being blocked before his/her requested lock is confirmed. The application case study shows that this concurrency control scheme has advantages of high responsiveness, unconstrained collaboration, and good data consistency maintenance.

## 1 Introduction

Real-time distributed collaborative editing system is an important branch of CSCW (Computer Supported Collaborative Work), and it is one of the most active research areas. It has the characteristics such as high responsiveness, high concurrency, and unconstrained collaboration. Therefore, the architecture of the system must be replicated or semi-replicated, which makes its concurrency control very difficult. Recently, a number of concurrency control algorithms for real-time collaborative editing systems have been proposed, e.g., traditional lock algorithm, tickle lock [1], floor control strategy [2], Undo/Redo model [3][4], and operational transformation [5][6][7]. However, all these algorithms have disadvantages in different aspects, and reading lock was not taken into account in them. Enlightened by operational transformation and optimistic locking mechanism, and with the consideration of both reading lock and writing lock, we propose a multi-granularity optimistic locking concurrency control scheme based on relative position, which has the characteristics of high responsiveness, unconstrained collaboration, and good data consistency maintenance. We have applied this scheme in our real-time collaborative editing system, and found that the scheme combines the advantages of traditional optimistic locking mechanism and operational transformation algorithm and meets the requirements of real-time collaborative editing systems well.

## 2 Optimistic Locking Scheme Based on Relative Position

### 2.1 Basic Concepts

In any centralized system, the server becomes the bottleneck. On the other hand, the concurrency control for full distributed system is very difficult. Therefore, the architecture of proposed collaborative editing system is semi-replicated, in which each user can create a new session or join an existing session. There are three roles in the proposed system: chief editor, co-editor and reader. The conflicts between locks are coordinated by the *chief editor* site in the session. Except the locking request, all editing operations of co-editors are sent to the other member sites in the same session. Additionally, in this system, the shared documents are replicated at the local storage of each cooperating site.

The basic terms referred to in this system are defined as follows:

**Definition 1:** *Insert*[ $S,P$ ] means inserting string  $S$  at position  $P$ .

**Definition 2:** *Delete*[ $N,P$ ] means deleting  $N$  characters starting from position  $P$ .

**Definition 3:** *Copy*[ $N,P$ ] means copying  $N$  characters starting from position  $P$ . It is only executed at the local site.

**Definition 4:** *Co-editor* is a user having joined a session.

**Definition 5:** *Chief editor* is the first user establishing the session, and she/he is responsible for the coordination of conflicts between locking requests in the session and the management of the session.

**Definition 6:** *Reader* can only read and copy the local replica but cannot edit it. Moreover, if she/he does not want his/her reading/copying region to be changed, she/he can put a reading lock on this region.

**Definition 7:** *Relative position* between two locks denotes the distance from the end position of the former lock to the start position of the later lock.

**Definition 8:** *Lock* is denoted by the following data structure: *lock* (*user\_id*, *type*, *t\_s*, *rel\_pos*, *s\_p*, *s\_l*, *n\_l*), where, *user\_id* denotes the ID of a lock's owner; *type* is used to identify that the lock is a reading lock or an editing lock; *t\_s* represents the timestamp when the lock is requested; *rel\_pos* denotes the relative position between the lock and its reference lock; *s\_p* denotes the absolute locking position of the lock; *s\_l* represents the length of the locking region when the lock is requested; and *n\_l*, which is changed dynamically with the editing operations going along in this lock, denotes the length of the locking region when the region is edited.

**Definition 9:** *Lock conflict* occurs when two or more concurrent exclusive locking operations overlap in their regions.

In the proposed scheme, locks are classified into reading locks and editing locks. All users cannot edit the locking region of reading locks, in which they can only read or copy. But the owner of editing lock can insert, delete, and copy in its locking region. In addition, we assume that one user can hold an editing lock and a reading lock at best at the same time, and reading locks are mutually inclusive, that is, two or more reading locks can overlap in their regions, but editing locks are mutually exclusive.

**Definition 10:** *Locking Table (LT)* is a list table recording the data of locks, and denoted by the following data structure: *locklist* (*lock<sub>0</sub>*, *lock<sub>1</sub>*, *lock<sub>2</sub>*, ..., *lock<sub>i</sub>*, ...). Each lock in *LT* is denoted by the data structure of Definition 8, and *lock<sub>0</sub>* is *lock-head*. Moreover, Locks in the *LT* are ordered by sort ascending of their start locking posi-

tion, and this order will be changed dynamically because of locking and unlocking operations. Furthermore, the largest lock number in *LT* is  $2n$ , here,  $n$  is the number of the co-editors in the same session.

**Definition 11:** *Lock-head* is the timestamp of the first lock, and it is the reference time when the first lock in the *LT* is being requested.

**Definition 12:** *Reference Lock* is the nearest to the requested lock and in front of it, and it is the reference for calculating the absolute start locking position of the requested lock. For example, it is assumed that the editing context is “ABCDEFGHIJKLMNQRST”, and there are three locks, as shown in Fig. 1. Here,  $Lock_1$  is the reference lock of  $Lock_2$ , and  $Lock_2$  is the reference lock of  $Lock_3$ . It is assumed that  $Lock_1[user1, w, lock_1, 0, 2, 4, 4]$  is included in the *LT*, and  $Lock_2$  is denoted by  $Lock_2[user2, w, 2, 5, ref\_lock[User1, w, Lock_1]]$  before being added into *LT*. While  $Lock_2$  is confirmed and added into *LT*, its absolute start locking position is calculated by the following equation:  $Lock_2.s\_p = Lock_1.s\_p + Lock_1.n\_l + Lock_2.rel\_pos$ . Then the absolute start locking position of  $Lock_2$  is:  $2 + 4 + 2 = 8$ , viz.,  $Lock_2[user2, w, Lock_1, 2, 8, 5, 5]$  is added into *LT*.

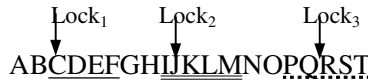


Fig. 1. A scenario of reference locks

**Definition 13:** *Undo Operation Set* includes all the editing operations going along in the locking region of an editing lock at a local site before the editing lock is confirmed.

It should be pointed out that the editing operations mentioned in this paper only include *Insert* and *Delete* operations, but not *Copy* operations.

## 2.2 Optimistic Locking Scheme Based on Relative Position

### 2.2.1 High Responsiveness

The high responsiveness of the proposed locking scheme is based on the notion of optimistic lock. When a user requests a lock on a region, she/he can edit the region without being blocked, and editing operations are saved into the *Undo Operation Set*. A requested lock will eventually become confirmed or aborted. When it is confirmed, its owner is then guaranteed to have an exclusive right to the region, and the operations in the *Undo Operation Set* go into effect and are sent to all remote sites in the same session, or else, these operations are undone and the owner of the lock will not be allowed to continue editing the region. Here, the period between the lock request and the lock confirmation/rejection is called the transaction period<sup>[8]</sup>. Editing operations of a local site going along during the transaction period are saved into the *Undo Operation Set*.

### 2.2.2 Consistency Maintenance for Locking Positions and Editing Positions

Since each cooperating site generates and broadcasts operations (including editing operations and locking operations) without synchronization, operations may be exe-

cuted at a position different from their natural position at remote sites. In order to solve this problem, we assume that each co-editor can only edit the locking region of her/his own editing lock, and the locking length of each editing lock is adjusted dynamically with the editing operations going along. In addition, we find that the relative position between two neighboring locks is fixed. Inspired by this idea, when a lock is requested or sent, its start locking position is denoted by the relative position between it and its reference lock. And the start position of editing operations in an editing lock is also denoted by the offset to the starting position of the lock. Then the position consistency of locking operations and editing operations are maintained.

**2.2.3 Conflict Coordination Scheme for Requested Locks**

Before editing, each co-editor must firstly apply for an editing lock on the region in which she/he wants to edit. But whether a reader applies for reading lock before reading or copying is determined by herself/himself, and she/he can read or copy at any position in the replica. If a requested reading lock is confirmed, the content in its region will not be changed before it is unlocked. At present, the contribution of reading lock for the concurrency control in collaborative editing system is still argued. We also only have done little research on it. Moreover, Conflicts between requested locks are coordinated by the *chief editor* site. If an editing lock requested is confirmed, its owner can edit or copy at any position in the locking region, and these editing operations, which are denoted by offset to the start locking position of the lock, are sent to other member sites in the same session. Then these operations are executed at the remote sites in turn respectively. If a co-editor wants to apply for a lock on a region, the local *LT* is scanned first in order to judge whether the lock conflicts with the others. If they conflict with each other, local editing operations in the region are prohibited, and the requested lock is aborted, otherwise, it is sent to the *chief editor* site. Before the requested lock is confirmed, the local user can edit the locking region in advance, and the operations going along during transaction period are saved into the *Undo Operation Set*. When the *chief editor* site receives the locking request, it also scans its *LT* to check whether the locking region of the lock conflicts with that of the other locks in the *LT* and whether its reference lock is out of date (here, reference lock out of date means that the reference lock is unlocked. And it brings in the result that

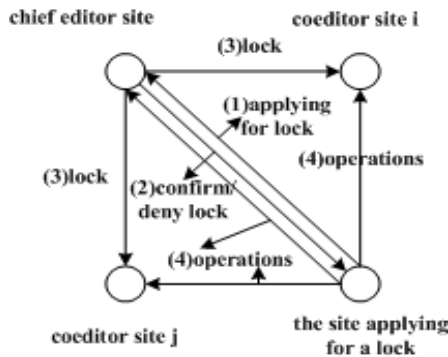


Fig. 2. A scenario of applying for locks and sending editing operations

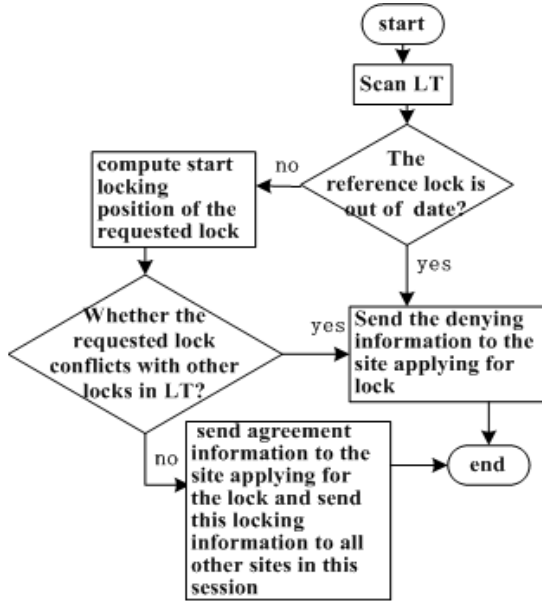


Fig. 3. The conflict coordination scheme for requested locks at chief editor site

locking position transformation of a requested lock lacks warrant or the transformation result violates its intention.). If not, it will reply with agreement information and send the information of the new confirmed lock to all member sites in the session, and the editing operations in the *Undo Operation Set* of the site applying for the lock take effect and the requested lock is confirmed, or else, the *chief editor* site will reply with denying information, and the requested lock is aborted. When the site applying for the lock receives the denying information, it will undo its editing operations done ahead of schedule according to its *Undo Operation Set*. The process of requesting a lock is shown in Fig. 2, and the conflict coordination scheme for requested lock at *chief editor* site is shown in Fig. 3.

On the contrary, if a user unlocks her/his lock, she/he must inform the *chief editor* site. Receiving this information, the *chief editor* site also undoes this lock and modifies its *LT*, then sends this information to other member sites in the session. When these member sites receive this information, they also unlock the lock and modify their *LT* correspondingly.

As mentioned above, we can see that if the locking region of a requested lock does not conflict with that of the others in the local *LT*, its owner can edit the region immediately without being blocked. Additionally, since awareness and other coordination mechanisms are implemented in our real-time collaborative editing system, it is rare that multi-co-editors apply for locks at a conflict position so that it is unusual to have undoing editing operations due to the abortion of requested lock. Therefore, a high responsiveness of the system is achieved.

### 3 Conflict Judgment and Locking Position Calculation Scheme

The site applying for a lock sends to the *chief editor* site the information of the locking request:  $apply\_lock((user\_id, lock\_tp, rel\_pos, lock\_len), ref\_lock(ref\_user\_id, ref\_tp, timestamp))$ . Here,  $user\_id$  denotes the ID of the user applying for the lock;  $lock\_tp$  denotes that the lock is an editing lock or a reading lock;  $rel\_pos$  represents the relative position between the requested lock and its reference lock;  $ref\_user\_id$  denotes the ID of the user holding its reference lock;  $ref\_tp$  denotes the type of the reference lock, and  $timestamp$  denotes the timestamp of the reference lock.

#### 3.1 Conflict Judgment and Locking Position Calculation for Requested Locks at Chief Editor Site

Because of the time delay of network and operation concurrency, when the information of a requested lock arrives at the *chief editor* site, and its reference lock is not out of date, there may be new locks held by other co-editors between the requested lock and its reference lock (it can be illustrated by some examples in Fig. 4.). Therefore, it must be judged whether the requested lock conflicts with these new locks according

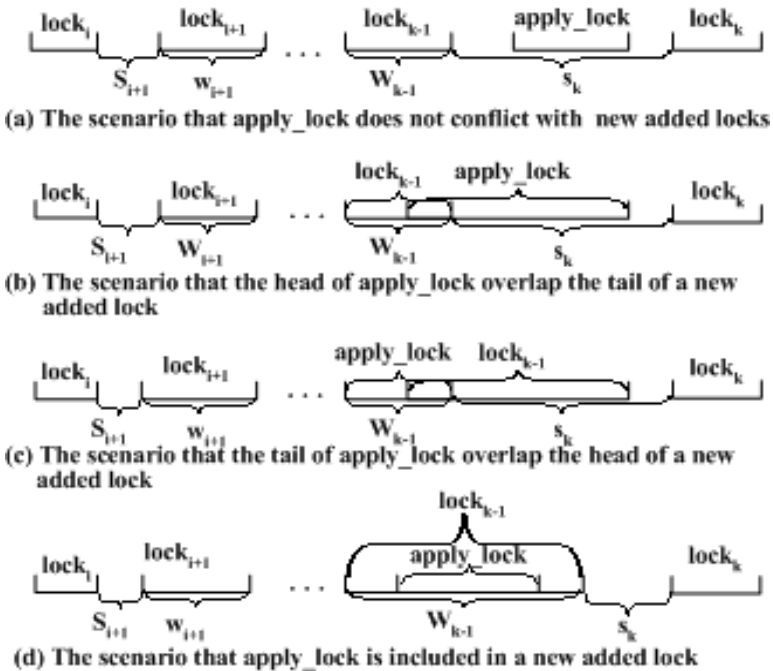


Fig. 4. Position relationships between  $apply\_lock$  and a new added lock

to their locking positions and the length of their locking region. If there is no conflict, the reference lock and the relative position between the requested lock and its new

reference lock are adjusted. In Fig. 4, *apply\_lock* denotes a requested lock; *lock<sub>i</sub>* denotes the reference lock of *apply\_lock*; *L* denotes the locking length of *apply\_lock*, and *r* denotes the relative position between *apply\_lock* and *lock<sub>i</sub>*; *s<sub>j</sub>* denotes the relative position between *lock<sub>j-1</sub>* and *lock<sub>j</sub>*, and *w<sub>j</sub>* denotes the locking length of *lock<sub>j</sub>*. The conflict judgment and locking position calculation methods in all possible cases are introduced respectively as follows:

- (1) *apply\_lock* and the locks between *lock<sub>i</sub>* and *apply\_lock* do not overlap in their regions, and it is shown in Fig. 4(a). In this case, there must be a *lock<sub>k</sub>* satisfying

$$r \geq \sum_{j=i+1}^{k-1} (s_j + w_j) \text{ and } r+L \leq (\sum_{j=i+1}^{k-1} (s_j + w_j) + s_k). \text{ We}$$

can see from Fig. 4 (a) that *apply\_lock* does not conflict with the other locks, but *lock<sub>k-1</sub>* becomes its new reference lock. Furthermore, for operations of other coeditors are going along in their own locking regions all the time, when the information of *apply\_lock* arrives at the *chief editor* site, editing operations in the locking regions of the locks between *apply\_lock* and *lock<sub>i</sub>* may have occurred. Accordingly, it will result in the change of the relative position *r* between *apply\_lock* and *lock<sub>i</sub>*. The new relative position *r'* can be calculated by the following equation:

$$r' = r + \sum_{m=i+1}^{k-1} (\text{lock}_{m,n\_l} - \text{lock}_{m,s\_l}) \tag{a}$$

Then the new relative position *apply\_lock.rel\_pos* between *apply\_lock* and its new reference lock *lock<sub>k-1</sub>* also can be calculated by the equation (b):

$$\text{apply\_lock.rel\_pos} = r' - \sum_{m=i+1}^{k-1} (\text{lock}_{m,rel\_pos} + \text{lock}_{m,n\_l}) \tag{b}$$

After the *apply\_lock.rel\_pos* is attained, the absolute start locking position of *apply\_lock* can be got by the following equation:

$$\text{apply\_lock.s\_p} = \text{lock}_{k-1,s\_p} + \text{lock}_{k-1,n\_l} + \text{apply\_lock.rel\_pos} \tag{c}$$

- (2) *apply\_lock* and the new locks between *apply\_lock* and *lock<sub>i</sub>* overlap in their locking regions, and it is shown in Fig. 4(b)(c)(d). In these cases, if both *lock<sub>k-1</sub>* and *apply\_lock* are reading locks, there is no conflict between them. But the reference lock of *apply\_lock* becomes *lock<sub>k-2</sub>*, and the relative position between *apply\_lock* and *lock<sub>k-2</sub>* can be attained with reference to the equation (a) and (b). On the other hand, if one of *apply\_lock* and *lock<sub>k-1</sub>* is a reading lock and the other is an editing one or both are editing locks, they conflict with each other. Then the *apply\_lock* is aborted, and the *chief editor* site will reply with the denying message to the site applying for *apply\_lock*.

In both conditions mentioned above, if there is no conflict, *apply\_lock* is confirmed, and the *chief editor* site adds the new information of *apply\_lock* (including its new reference lock, new relative position and new absolute start locking position) into the *chief editor* site's *LT* and sends this information to all member sites in the same session.

### 3.2 Locking Position Calculation of a New Added Lock at All Member Sites

Because of the operation concurrency, when the information of an added lock sent by the *chief editor* site arrives at co-editor sites, a local user may have applied for new locks between the new added lock and its reference lock but not received the agreement information from the *chief editor* site. It should be pointed out that the local site is only able to apply for one editing lock and one reading lock between the new added lock and its reference lock at best or one of them, which is shown in Fig. 5. In this figure, *add\_lock* denotes the new added lock sent by the *chief editor* site, *refer\_lock* denotes the reference lock of *add\_lock*, and *local\_lock* represents the lock which is requested by the local user between *add\_lock* and *refer\_lock*, but is not confirmed by the *chief editor* site. In this process, one of the following two cases will emerge, (1) If *add\_lock* and the new locks between *add\_lock* and *refer\_lock* do not overlap in their regions; or if *add\_lock* and local locks are all reading locks, even in the condition that they overlap in their regions, there is no conflict. This case is illustrated in Fig. 5 (a) and (b). In this case, the reference lock of *add\_lock* and the relative position between *add\_lock* and its reference lock are also adjusted according to the scheme mentioned in Section 3.1; (2) If *add\_lock* and local locks overlap in their regions, and at the same time, both *add\_lock* and the local lock, the locking region of which overlap *add\_lock*, are editing locks or one is an editing lock and the other is a reading lock, these two locks conflict with each other, and the local lock conflicting with *add\_lock* is aborted. This case is illustrated in Fig.5 (c) and (d). Correspondingly, the parameters of *add\_lock* are also adjusted according to the scheme mentioned in Section 3.1. Then *add\_lock* is added into the local *LT*.

As mentioned above, we can see that, in the case that there is no conflict between the requested lock and the locks in *LT* and that its reference lock is not out of date, if

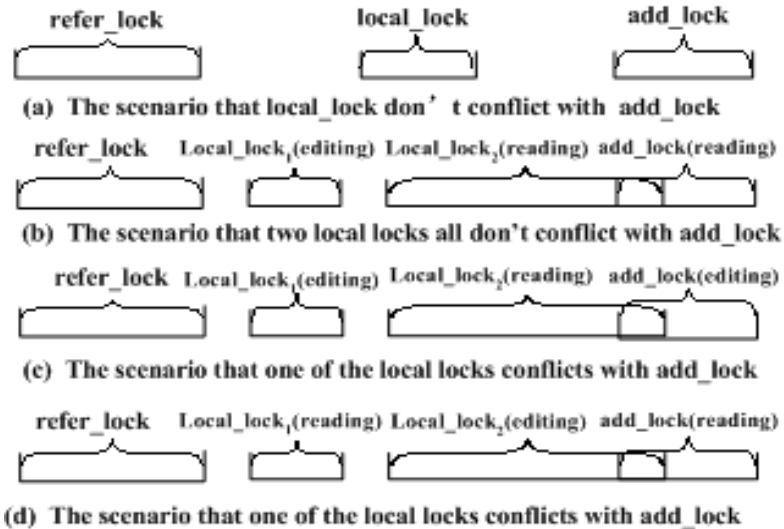


Fig. 5. Position relationships between a new added lock and new requested locks in a local site



co-editors edit the locking region of their own editing lock respectively, the area out of their locking region will not be changed. Therefore, the relative position between two neighboring locks will not be changed too. Additionally, during a lock being requested, if there are new locks requested by other co-editors between the requested lock and its reference lock, the locking position and the reference lock of the requested lock will be adjusted according to the scheme mentioned in Section 3.1. On the other hand, at the *chief editor* site, if a requested lock conflicts with the locks in *LT* or its reference lock is out of date, the requested lock is aborted, and editing operations in it done ahead of schedule are canceled according to *undo operation set*. Then its locking position consistency at *chief editor* site is guaranteed. Moreover, when the information of a new added lock arrives at co-editor sites, if a local site has applied for new locks between the new added lock and its reference lock, and the local user has edited in it, the reference lock and the locking position are also adjusted according to the scheme mentioned in Section 3.1. By this means, the locking position consistency at each cooperative site can be maintained. Of course, if the local co-editor did not apply for a new lock between the new added lock and its reference lock, the parameters of this new added lock will not be changed.

### 4 Application of the Scheme in Collaborative Editing System

In our real-time distributed collaborative editing system, the communication architecture between cooperative sites is fully distributed except applying for lock, and our concurrency control scheme has adopted in this system. A cooperative editing scenario including three co-editors is shown in Fig. 6, where, the local user holds an editing lock lock<sub>2</sub> [336,213], and the other two co-editors hold a reading lock lock<sub>1</sub>

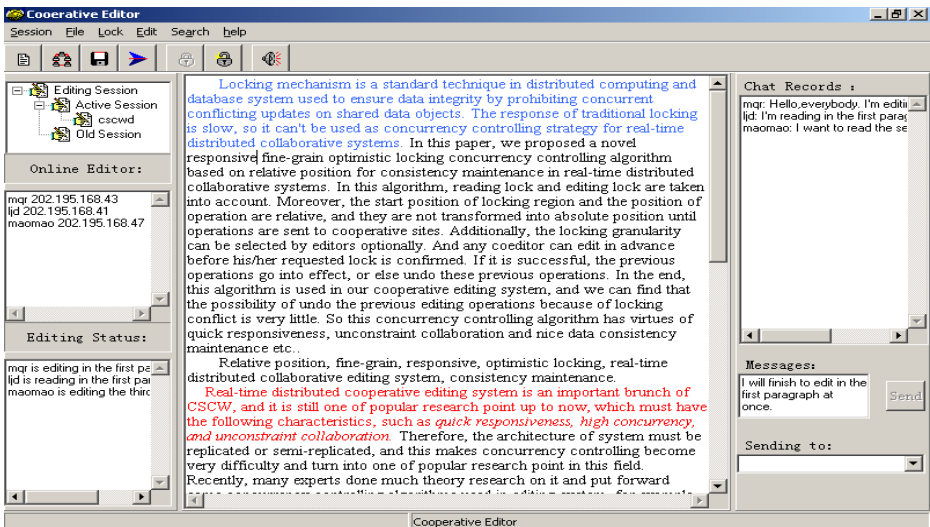


Fig. 6. A scenario of the scheme application in the real-time collaborative editing system

[1,336] and an editing lock  $lock_3$  [1558,264] respectively. In our system, the locking regions of different co-editors are distinguished by different colors. The application result shows that our concurrency control scheme can maintain the consistency of the replica at each cooperative site, viz., it can ensure the CCI (Convergence, Causality-preservation, and Intention-preservation) properties of consistency model<sup>[9]</sup>.

## 5 Comparisons with Related Work

In the current collaborative editing system, the tickle lock is adopted. Since the granularity of lock in this algorithm is one section at least, multi-co-editors cannot edit in the same section at the same time<sup>[1]</sup>. Whereas, in our scheme, the granularity of locks can be selected by co-editors optionally.

The floor control algorithm is used in the cooperative editing of JCE, in which each co-editor must get floor before editing, so it is fit for discussion, instead of the real-time unconstrained collaborative editing. However, our scheme can guarantee the high responsiveness and unconstrained collaboration.

In addition, high responsiveness, convergence and unconstrained collaboration can be achieved in the operational transformation, but it cannot maintain the semantic consistency in context. So the locking mechanism is also integrated into this scheme in its later work<sup>[9]</sup>. Not only having the advantages of operational transformation, but also our scheme is able to guarantee the semantic consistency in context, and takes reading lock into account.

## 6 Conclusion

In this paper, we proposed a novel multi-granularity optimistic locking scheme based on relative position for real-time collaborative editing systems. We start from pointing out that locking is able to help maintain semantic consistency but the consistency of locking positions is difficult to maintain. Based on this observation and on the fact that the relative position between two neighboring locks will not be changed, the locking positions and operation positions are denoted by relative positions when they are sent. Then, the schemes of conflict judgment between two locks and locking position calculation at the *chief editor* site and other member sites have been proposed. At the same time, the reading lock has been taken into account in the proposed approach. The similarities and differences between the proposed scheme and other existing concurrency control schemes in collaborative editing systems are also discussed.

The proposed scheme has been implemented in our real-time collaborative editing system. By means of this prototype system, we have carried a usability study on the proposed scheme from end-users' perspective. The result shows that the features of high responsiveness, unconstrained collaboration and high concurrency are achieved. Moreover, after a further improvement, our scheme can be used for the concurrency control of other distributed shared objects such as linear list.

## Acknowledgements

The work presented in this paper is supported by National Natural Science Foundation of China under Grant No. 60273040; Qinglan Project of Jiangsu Province No. 1191170004; the Youth Science Foundation of Jiangsu University No. 1241170005.

## References

1. Greif, I., Seliger, R., and Wehl, W.: A Case Study of CES: A Distributed Collaborative Editing System Implemented in Argus. *IEEE Transaction on Software Engineering*, 18(9) (1992) 827-839
2. Abde-Wahab, H., Kvande, B., Kim, O., and Pavreau, J.P.: An Internet Collaborative Environment for Shared Java Applications. *Proc. of 5th IEEE Computer Society Workshop on Future Trends of Distributed Computing Systems (FTDCS'97)*. IEEE Computer Society Press, Los Alamitos, (1997) 112-117
3. Choudhary, R. and Dewan, P.: A General Multi-User Undo/Redo Model. In: *Proc. of European Conference on Computer Supported Work*. Kluwer Academic Publishers, Dordrecht, (1995) 231-246
4. Sun, C.Z.: Undo Any Operation at Any Time in Group Editor. *Proc. of ACM Conference on Computer Supported Cooperative Work*. ACM Press, New York, (2000) 191-200
5. Ellis, C.A., Gibbs, S.J., and Rein, G.L.: Groupware: some issues and experiences. *Communications of the ACM*, 34(1) (1991) 39-58
6. Sun, C.Z., Jia, X., Zhang, Y., and Yang, Y.: A Generation Transformation Scheme for Consistency Maintenance in Real-time Cooperative Editing Systems. *Proc. of International ACM SIGGROUP Conference on Supporting Group Work*. ACM Press, New York, (1997) 425-434
7. Sun, C.Z. and Ellis, C.A.: Operational Transformation in Real-Time Group Editors: Issues, Algorithms, and Achievements. *Proc. of the ACM Conference on CSCW*, ACM Press, New York, (1998) 59-68
8. Sun, C.Z.: Optimal and Responsive Fine-Grain Locking in Internet-Based Collaborative System. *IEEE Transaction on Parallel and Distributed Systems*, 13(9) (2002) 994-1008
9. Sun, C.Z. and Sobic, R.: Optional Locking Integrated with Operational Transformation in Distributed Real-time Group Editors. *Proc. of ACM 18th Symposium on Principles of Distributed Computing*, ACM Press, New York, (1999) 43-52

# Research on Content-Based Text Retrieval and Collaborative Filtering in Hybrid Peer-to-Peer Networks

Shaozi Li<sup>1,2</sup>, Changle Zhou<sup>2</sup>, and Huowang Chen<sup>1</sup>

<sup>1</sup> School of Computer Science, National University of Defense Technology, Changsha, P.R. China, 410073

<sup>2</sup> Department of Computer Science, Xiamen University, Xiamen, P.R. China, 361005  
szlig@xmu.edu.cn

**Abstract.** Hybrid peer-to-peer architectures use special nodes to provide directory services for regions of the network (“regional directory services”). They are a potentially powerful model for developing large-scale networks of complex digital libraries. This paper presents our recent research work on the new content-based text filtering and collaborative filtering based on hybrid P2P (Peer-to-Peer) networks. From various perspectives, our work focuses on how to share the text content and recommend information based on hybrid P2P networks. Several models are proposed to implement the content-based text retrieval and collaborative filtering effectively. These models are then evaluated and validated through implementations and analyses. The results show some advantages of the proposed approach for the content-based filtering algorithm based on lexical chain and collaborative filtering algorithm in hybrid P2P network and potential applications in complex digital libraries and distributed information sharing.

## 1 Introduction

Peer-to-Peer (P2P) computing is a relatively new approach to federated search of large networks of digital libraries. In P2P networks, the nodes can send and receive information as both servers and clients. *Pure* P2P architectures are completely decentralized; each node can issue requests which can be satisfied, or route requests to other nodes. *Hybrid* P2P architectures include two types of nodes: *leaf nodes* and *directory nodes*. *Leaf nodes* provide information as well as post requests (“queries”). *Leaf nodes* can be used to model an individual with an information need or an information resource (e.g., a digital library). *Directory nodes* do not have contents of their own but provide regionally centralized directory services for the network to improve the routing of information requests. Directory nodes are also called “ultrapeers”, “hubs”, or “supernodes” in the research literature. Each directory node provides directory services for portions of the network and directory nodes work in a cooperative manner to cover the whole network.

Early P2P architectures provided federated search by either relying on a single centralized directory service or employing the *flooding* technique in completely decentralized manner (a node broadcasting query messages to all of its neighbors) to

decide how to route query messages. The former approach suffers from a single point of failure and has limited scalability, while the latter approach is less efficient and may overload the network. Hybrid P2P architectures that use multiple decentralized directory services were developed to solve these problems.

Although research on information systems using P2P architectures is very active recently, most recent research focuses on improving the efficiency, robustness, and load-balancing of distributed information storage or file-sharing systems [1-4]. Document retrieval in P2P networks has so far mostly been limited to simple *keyword-based* methods: Matches between query terms and the keywords including in documents. These techniques may be sufficient for networks of small digital libraries that use well-known naming conventions and provide simple services. But some times, it is not available because different words can present the same things, for example, the Chinese word ‘计算机’ and ‘电脑’ have the same meaning ‘Computer’. So concept based information retrieval is important.

This paper proposes a new content-based text retrieval and collaborative filtering model based on P2P network, the model includes the lexical-chain based text filtering algorithm and P2P architecture based information recommendation module. It presents the basic technique and method of implementing collaborative research and information sharing on this module, describes the technique of content-based information filtering and the algorithm of information recommending. The experimental results on content-based filtering algorithm are presented and the future works are discussed.

## 2 Content-Based Text Retrieval Model in Hybrid P2P Network

In our content-based texts retrieval model in hybrid P2P networks, the leaf nodes provide request and documents, and the directory nodes do not have documents but provides regionally centralized directory services for the network to improve the routing of information requests. Each directory node provides directory services for portions of the network and directory nodes work in a cooperative manner to cover the whole network.

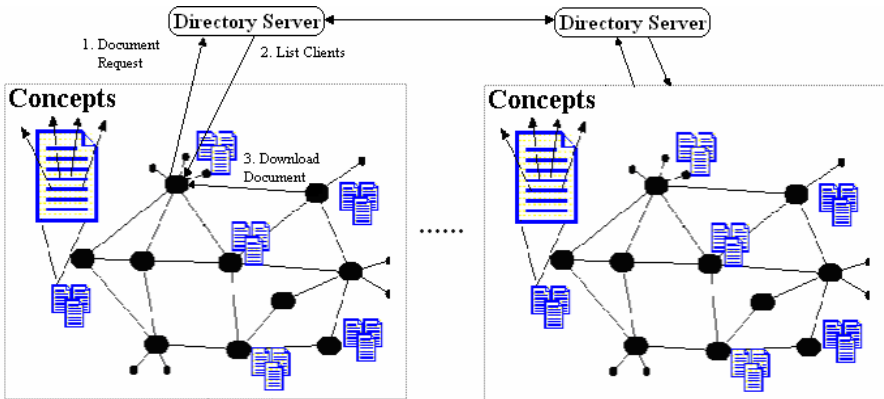


Fig. 1. The content-based documents retrieval model in hybrid P2P network

While the leaf node requires the document, it can post the request to the directory node of the same region. The directory node routes the query to other peer of this region. Every other peer will compare the query with the documents based on concepts, calculate the relativity between them. Then, all peers will respond to the directory server. After that, the directory server can rank all responses and send the list clients with the required contents to the leaf node. When the leaf node gets the list peers, it will connect the peer with required document, and download the document directly from this peer.

If all peers in this region do not have the document required, the directory will post the query to another directory node and wait the response from it.

Fig. 1 shows the content-based documents retrieval model in hybrid P2P network.

### 3 The Content-Based Filtering Algorithm Based on Lexical Chain

#### 3.1 About Lexical Chain

As we known, an article or a paragraph is not simply enumerating some random words, but constituted mainly by a series of words or phrases that express synonymous. We can call this phenomenon cohesion, which makes these words and phrases concatenated according to some certain grammar and expression.

A lexical chain is a series of words constituted by some adjacent words under the same topic. These words appear simultaneously in the same lexical environment, as they are expressing the same thing.

E.g.: A paragraph about economic development: Efforts to develop the *economy* lacked a unified, focused approach. Basic requirements for sustained development had been established in only one *sector*, agriculture. Lack of well-developed *economic system*, puts the *economy* at greater risk to changing external *economic* factors.

From this paragraph, we can find such a lexical chain {economy, sector, economic system, economy, economic}. These related words and phrases appear sequentially under the topic economy. Distances exist between these words, and their simultaneous appearance is within a certain scope in a concrete article, but they are not limited in sentences. Lexical chain describes semantic units in texts, and does not need the complicated natural language process while being built.

Morris and Hirst [5] first brought in the concept of lexical chain, and used it in text segmenting, so as to acquire a text structure. The basic thought is: as the lexical chain is constituted by a series of relevant words that express the same thing or meaning, if we find out these chains, then we get the text structure, moreover, different chains constitute segments of this text. Yet because of the limitation of that time, this algorithm did not be realized.

Lexical chain is now used in information extraction, information retrieval, checking unsuitable words in a text, analyzing and computing texts' similarity, constructing automatic links for super texts, segmenting texts, words disambiguating etc. [5-8].

#### 3.2 The Way to Form a Lexical Chain

While building a lexical chain to express a text, we should firstly consider how to choose words. It means which words are suitable to be candidate words. After analyz-

ing a text, we delete all empty words: pronouns, modal verb, prepositions or adverbs with a subordinate clause and articles. Some other words that appear frequently, such as good, do, taking etc., we also put them into the table that includes the unused words. The rest are all candidates.

The next problem is the word relationship. We build lexical chain according to the word relationship in dictionary. We use WordNet [9] as the source knowledge, which was designed Miller and Beckwith of Cognitive Science Laboratory, Princeton University, USA. As an online lexical consulting system (an English word database that can be read by machines online), WordNet is a machine dictionary based on psycholinguistic principles. It uses our familiar spellings to express morphology and the synonym set Synsets (a list of synonyms which can be substituted each other according to certain context) to express the meaning of a word. So far, WordNet embodies about 95600 lemmas, including 51500 words and 44100 compound words. They are organized roughly 70100 acceptations or synonym sets, describing relationships such as hyponymy, synonymous, antonymous, meronym / holonym etc.

Here we consider 3 relationships: extra-strong, strong, and medium-strong.

a) Extra-strong:

It means the repetition of two words, neglecting their distance.

b) Strong:

It can be considered in 3 situations (set a window between two words, usually 7 sentences):

b1) Both the two appear in the same synonym set, e.g. human and person appear in the same set {person, individual, someone, man, mortal, human, soul}

b2) There is some kind of semantic relationship between some certain sets in each synonym set of the two words, e.g. one synonym set of precursor {predecessor, precursor, antecedent } has a antonymous relationship with successor's synonym set {successor}. We call this relationship horizontal connecting.

b3) If one is a phrase or compound word, and some words in this phrase or compound word appear in other synonym sets, then we do not consider what kind

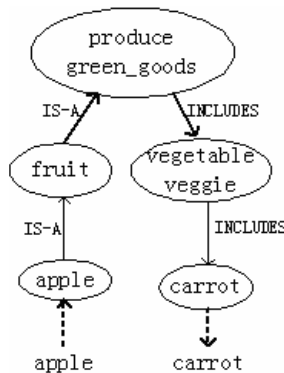


Fig. 2. The medium-strong path between apple and carrot

of relationship this containment is, e.g., private\_school. The synonym set{ private\_school } is included by the synonym set{school}. Although there is a relation of hyponymy, we consider school and private\_school as strong.

c) Medium-strong:

There is a path connecting two words (define the distance between two words as a window, usually 3 sentences) and we limit the path length between 2 and 5. Figure 2 shows the medium-strong path between apple and carrot.

According to the relationship between key words, the formula for computing a weight value is described as the formula (1).

$$\text{Weight}_{ij} = \begin{cases} 3C & \text{Extra-strong} \\ 2C & \text{Strong} \\ C - \|\text{path}_{ij}\| - \beta * (\#\text{turn}) & \text{Medium-strong} \\ 0 & \text{Others} \end{cases} \quad (1)$$

In the formula (1), C,  $\beta$  are constants,  $\text{path}_{ij}$  is the semantic distance between two words, #turn means how many times directions change in the path.

The basic idea is as follows: For each word  $W_i$  in the text, if  $W_i$  is a candidate word and there is some relationship between  $W_i$  and the lexical-chain (the interval between them cannot be too big), then calculate the power value between  $W_i$  and the lexical-chain and append  $W_i$  into the lexical-chain.

### 3.3 Expressing Text

Traditional information filtering technique primarily filters out relevant texts by key words searching and statistics, just respectively computes the frequency of each key word in a text, but neglects the relationship between key words or key words and the topic. This technique isolating key words has many defects. Here we analyze three main factors that caused these defects in traditional key word technique and then give the corresponding solutions:

#### 3.3.1 Influence of the Useless Words

That means characteristics that can appear in the all categories cannot represent a category. Some of these words belong to the set that includes the unused word, and others carry little information, so we will delete them.

#### 3.3.2 Influence of the Relationship Between Words

It can be divided into two instances: one is because of synonyms (e.g. Chinese word “计算机” and “电脑”), while the other is because of certain semantic relationship between words. In the medical category, for example, certain relationship exists in “doctor”, “nurse”, “hospital”, “sickbed”, “operating room”, “diagnosis”, “infection”, “state of an illness”, “antibody” etc. The existence of one characteristic can replace others to some extent. The frequency that each characteristic appears respectively may be little or overlaid by some irrelevant words of high frequency. As the above distance formula does not take account of such influence while computing, it will also



lead to inaccuracy in distance computing. For this reason, we should take the semantic relation between words account. If they express the same topic, then their semantic distance in a dictionary is short, and they can be put together automatically during the process of text analyzing and considered comprehensively while computing the similar degree.

E.g.: Information of some words from a text as follows: {{information: 3, technique: 1, Bayesian-technique: 1, datum: 2, model: 1, area: 1} {computer: 4}}, the figure after each word means the times this word appeared in the text.

If we only consider the frequency of each word, computer's frequency is the highest. Yet we find other words are related strongly in semantics and can complement each other, so each one's importance is elevated in this way.

### 3.3.3 Influence of Unequal Status Between Words

Although how important each key word is supportive to the topic can be shown by the times that it appeared, it is not enough. Often we need not read the whole passage while reading, but can find out the topic accurately from the title or the first paragraph. This means there are some characteristic words supporting certain topic strongly (decision characteristics), their existences decide the topic to a great extent. Yet in a vector space model, this kind of decision may be submerged by the influence of numerous non-decision characteristics. In that case, we bring in the concept of characteristic district.

Text characteristic district is a district that can show a text's topic, including headlines, abstracts, key words and references. Yet not all texts contain an abstract, key words or references, so we let these structure units alternative. Some Chinese researchers found by sample statistics that the coherent degree of natural science thesis' headlines and contents is 98% in local Chinese periodicals, and the degree in news texts is 91%. Almost every article contains a headline, for this reason, a headline is one main text characteristic.

Clue words are those summarizing or generalizing words, such as "anyway", "in a word", "to sum up" etc. We will enhance the importance of the words included in characteristic districts, at the same time, intensify the power of the words after clue words to enhance their importance.

According to above, we will use lexical chain to express user templates and unknown texts. Firstly, we analyze texts handed over by user, build lexical chain to express them, then build user template which automatically learns during filtering to express users' interest and meet their interest better. As for unknown texts, we adopt the same method to build lexical chain for expressing them.

## 3.4 Analyzing Text

However, not all words in a text can be used for building lexical chain. Only key words that express the meaning of a text most clearly can be used. Methods for expressing texts are as follows:

**Preprocessing Text:** Withdraw etyma and recognize phrases etc.

**Part-of-Speech Tagging:** Part-of-Speech Tagging for words in the text.

**Extracting Key Words:** Get rid of the following words in texts: articles(e.g. a, an, the), preposition or adverb with a subordinate clause (e.g. to, of, in), modal verb(e.g. would, must), and conjunction(e.g. and) etc.. Let's make a definition  $W(s,w,c)$ , in which  $w$  means a word,  $s$  means the word's sequence in a text, and  $c$  means the part of speech. For example, (12,think, verb) means the 12th word in  $W$  is 'think' and its part of speech is verb. We also can evaluate different powers for different parts of speech to show their importance, and nouns are usually most important. As for those ones appeared in headlines, the first or last paragraph, or at the beginning or end of a paragraph, we can also enhance their power. In addition, we can set a valve value to get rid of words which frequency under it.

**Expressing texts with lexical chain:** Now, we get series of words. After lexical chain built automatically, we will get the text's lexical-chain expression.

### 3.5 Filtering Text

So far, texts and users' interest are expressed by lexical chain. The relevant degree between texts and users' interest can be estimated by the cosine value of the formula :

$$\cos(a) = \frac{\sum_{ij} V_i * T_j}{\sqrt{\sum_i V_i^2 + \sum_i T_i^2}} \quad (2)$$

Thereinto,  $V=\{V_1, V_2, \dots, V_n\}$  is a vector expression of a text's lexical chain,  $T=\{T_1, T_2, \dots, T_n\}$  is a vector expression of lexical chain of users' interest. The less  $\alpha$  is, the closer these texts relate to users' interest.

Among all filtered texts, we can make an order on relevant degree to feed back users according to this value, or we can set a valve value  $k$ , if the relevant degree between texts and users' interest above  $k$ , then we consider these texts meet users' interest, and return the ordered texts to users according to relevant degree, finally, get rid of all rest texts which under the value or store them somewhere for users to deal with when free. We can take users' feedback into account, if almost every text we filtered out is considered interesting to users, then we reduce  $k$ , whereas, increase  $k$ .

## 4 Collaborative Filtering Algorithm

There are many different techniques for implementing recommender systems [10]. Collaborative filtering is the most successful recommender system technology to date. The main idea of collaborative filtering is to recommend new documents to users according to how similar their tastes are to other users'. If two users tend to agree on what they like, the system will recommend the same documents to them. Therefore, collaborative filtering enables people to get the useful information with little effort by leveraging others' efforts.

In general, collaborative filtering is a three stage process of finding similar users (neighbors), computing predicated ratings, and applying the predictions as recommendations to the user [11].

To generate predictions for a user, the system first identifies this user's "neighbors", other users whose interests correlate highly to the user's. There are several possible options for the correlation coefficient, and the most common one is the constrained Pearson correlation:

$$\text{correl}(u_i, u_j) = \frac{(u_i - z)^T (u_j - z)}{\|u_i - z\| \|u_j - z\|}, \tag{3}$$

where  $u_i$  and  $u_j$  are two users' ratings vectors, and  $z$  is the neutral rating which is subtracted from the vector. For simplicity, we assume that  $z$  is subtracted off in the original ratings matrix, although in practice  $z$  must be taken into account when presenting a recommendation to the user. Now, equation (3) becomes (4) as follows:

$$\text{correl}(u_i, u_j) = \frac{u_i^T u_j}{\|u_i\| \|u_j\|}, \tag{4}$$

which is also known as the cosine similarity measure in information retrieval. In stage two, the system generates predictions for the user by computing a weighted average of each neighbor's rating scaled by their correlation value,

$$r_{i,j} = \frac{1}{|N(u_i)|} \sum_{u_k \in N(u_i)} \text{correl}(u_i, u_k) * r_{k,j} \tag{5}$$

Using this formula, we can do prediction for users' rating. With more rating information, we may be able to make a better recommendation for users in the future.

The third stage, application of the prediction, involves adding  $z$  back to the prediction and presenting the prediction to the user as a recommendation if it is high enough. If the user follows some of the recommendations and provides feedback to correct the predictions, the system learns the user's tastes and his relationship to the community over time.

## 5 Experimental Results and Analysis of the Content-Based Filtering Model

Two main criteria for evaluating text filtering systems are precision and recall. If the system has been informed of users' interest, the whole text will be logically divided into four parts after filtered: relevant or irrelevant texts that have been filtered out; relevant or irrelevant texts that have not been filtered out.

$$\text{Precision} = \frac{\text{The relevant texts that have been filtered out}}{\text{The total texts that have been filtered out}} \tag{6}$$

$$\text{Recall} = \frac{\text{The relevant texts that have been filtered out}}{\text{The relevant texts number of text set}} \tag{7}$$

The efficiency of a text filtering systems is generally described by average precision, and the visual explanation is the area of the precision/recall curve line.

This paper uses the medical corpus OHSUMED on TREC-9, which is a subset of MEDLINE in the famous National Library of Medicine and constituted by the medical literatures from 1988 to 1991, including 348, 566 texts from 270 medical periodicals, with a content of 400MB. In addition, literatures in 1987 are used as training corpus, while literatures between 1988 and 1991 as testing corpus [12]. Our comparative experiments show as the following table.

**Table 1.** The experimental results

	The traditional key-word based filtering system	The lexical chain based filtering system	Difference
Average precision	38.53%	47.46%	8.93%

In traditional vector space models, since each key word is considered respectively, we can only use the times that a key word appeared but not make full use of the information how words are correlated in a text. However, the precision will be improved if the lexical chain is used.

## 6 Conclusion and Future Work

In this paper, we proposed a model which combines content-based text filtering with collaborative filtering for the information sharing based on peer-to-peer and introduced a new content-based filtering algorithm based on lexical chain. We proposed several models to implement the content-based text retrieval and collaborative filtering effectively. These models are then evaluated and validated through implementations and analyses. The results show some advantages of the proposed approach for the content-based filtering algorithm based on lexical chain and collaborative filtering algorithm in hybrid P2P network and potential applications in complex digital libraries and distributed information sharing. We are working on the implementation of the collaborative scientific research system with our hybrid filtering model based on peer-to-peer and improve the efficiency of both content-based filtering and recommendation processes.

## Acknowledgement

This project is supported by The Natural Science Foundation of Fujian Province (Project Number: A0310009), Science & Technology of Fujian Province (Project Number:2001J005), Academician Fund of Xiamen University, China. Special thanks to Yang Cao, Dazhen Lin and Dandan Liu for their contributions to the related research work and preparation of this paper.

## References

1. Aberer, K.: P-Grid: A self-organizing access structure for P2P information systems. In Proc. of the 6th International Conference on Cooperative Information Systems, (2001).
2. Dabek, F., Kaashoek, M., Karger, D., Morris, R., Stoica, I.: Wide-area cooperative storage with CFS. In Proc. of the 18th ACM Symposium on Operating Systems Principles, (2001).
3. Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Shenker, S.: A scalable content-addressable network. In Proc. of the ACM SIGCOMM'01 Conference, (2001)161-172
4. Tang, C., Xu, Z. Mahalingam, M.: PeerSearch: Efficient information retrieval in peer-to-peer networks. In Proc. of HotNets-I, ACM SIGCOMM, (2002) 1-6
5. Morris, J., Hirst, G.: Lexical Cohesion Computed by Thesaural relations as an Indicator of the Structure of Text. *Computational Linguistics*, 17(1) (1991) 21-48
6. Silber, G., McCoy K.: Efficiently Computed Lexical Chains as an Intermediate Representation for Automatic Text Summarization. *Computational Linguistics*, 28(1) (2002) 1-11
7. Silber, H. G., McCoy, K.: Efficient Text Summarization Using Lexical Chains. In Proceedings of the 13th International Conference on Intelligent User Interfaces, IUI2000, New Orleans, (2000) 252-255
8. Stephen, G.: Lexical Semantics and Automatic Hypertext Construction. In: *ACM Computing Surveys* 31(4) (1999) 22-26
9. Miller, G.A., etc.: WordNet-a lexical database for the English language. <http://wordnet.princeton.edu/>
10. Terveen, L., Hill, W.: Human Computer Collaboration in Recommender Systems. In J. Carroll (Ed.), *Human Computer Interaction in the New Millennium*, Addison-Wesley, New York, (2001)
11. Soboroff, I., Nicholas, C.: Collaborative Filtering and the Generalized Vector Space Model. In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Athens, Greece, (2000) 351-353
12. Robertson, S., Hull, D.: The TREC-9 Filtering Track Final Report, Proceeding of the Ninth Text Retrieval Conference(TREC-9), (2001)

# On the Stochastic Overlay Simulation Network

Ke-Jian Liu<sup>1</sup>, Zhen-Wei Yu<sup>2</sup>, and Zhong-Qing Cheng<sup>3</sup>

<sup>1</sup> Network Information Center, Chinese People's Public Security University,  
Beijing, 100038, China  
lkj\_sl16@sina.com

<sup>2</sup> Graduate Student College, China University of Mining and Technology,  
Beijing, 100083, China  
zwyu@cumtb.edu.cn

<sup>3</sup> Department of Logistic Command & Engineering, Naval University of Engineering,  
Tianjin, 300450, China  
bace@tom.com

**Abstract.** This paper proposes a method of generating stochastic overlay simulation network with exact average degree (EAD). It discusses regional distribution of stochastic nodes, selection of core nodes and overlay functional nodes, deductions of new formula of the connectivity probability and the convergence of the connection probability, classification strategy of increasing degree, fast connection strategy of stochastic network. Further explanation of the performance of the stochastic overlay simulation network is also presented.

## 1 Introduction

The research on network related issues is always combined with simulation and test. It is difficult to carry out test in the real network at present; meanwhile, test carried out in a specific network will restrict the result to the specific experimental network. A protocol, algorithm or strategy developed and performed well in a specific network may not work well in another network or even in the same network if the network topology is changed greatly.

As present, simulation of specific network (e.g., stochastic network) is based on whole network. It is assumed that all the nodes in real network bear CSCWD functions. As a result, it underestimates actual network characteristics <sup>[1]</sup>, and the corresponding simulation results have little practical meaning.

The related simulation or test will be more instructional if the stochastic network simulation model further approaches to real network in the aspect of main performances. A good stochastic network simulation model is not only the platform of fundamental research on network architecture but also the foundation of simulation of protocols, algorithms and all kinds of overlay network related applications.

## 2 The Necessity of Research on EAD Simulation Model

Waxman proposed two kinds of models to generate stochastic networks: RG1 and RG2<sup>[2]</sup>. The network nodes generated by RG1 randomly are distributed in rectangular

grids, and the coordinates of nodes are stochastic integers in consistent distribution, and the distance between every two nodes is a Euclid length. As is different from model RG1, for the network nodes generated by RG2, the distance between every two nodes is a random value between (0, L) with uniform distribution. In both models, there is a certain probability determining whether to connect the two nodes. The probability is determined by distance between the two nodes. The probability of connecting nodes  $u$  and  $v$  is determined by formula (1):

$$p(u, v) = \beta e^{-\frac{d(u,v)}{\alpha L}} \tag{1}$$

A random value  $R_d$  with uniform distribution is generated between 0 and  $R_{MAX}$ , and the connection between nodes  $u$  and  $v$  exists if and only if:  $p(u, v) \geq \frac{R_d}{R_{MAX}}$

In formula (1),  $d(u,v)$  is the distance between nodes  $u$  and  $v$ ,  $L$  is the longest distance between nodes, and parameters  $\alpha$  and  $\beta$  determine the features of network graph, the values of which are in (0,1). The bigger  $\beta$  is, the bigger the average degree; and the bigger  $\alpha$  is, the bigger the ratio of long edge to short edge.

In fact, we can prove that the average degree of stochastic networks in Waxman’s model is not convergent as follows. So it is impossible to simulate large-scale actual network.

From the formula (1) we have the degree of node  $u$  such as:  $\mu = \sum_{i=1}^{n-1} f(u, v_i)$

where  $v_i \in V, u \neq v_i$

and the function  $f(u, v)$  is as follows :  $f(u, v) = \begin{cases} 1 & p(u, v) \geq rand(0,1) \\ 0 & p(u, v) < rand(0,1) \end{cases}$

The expectation of  $\mu$  is as following:

$$E(\mu) = E\left(\sum_{i=1}^{n-1} f(u, v_i)\right) = \sum_{i=1}^{n-1} E(f(u, v_i)) \tag{2}$$

Let  $p(u, v) = \beta e^{-\frac{1}{\alpha}}$  (minimum of  $p(u, v)$ ), we have

$$f_i(u, v) = \begin{cases} 1 & rand(0,1) \leq \beta \exp(-1/\alpha) \\ 0 & rand(0,1) > \beta \exp(-1/\alpha) \end{cases} \tag{3}$$

If formula (3) is substituted into (2), we have the expectation of node.

$$E(\mu) \geq \sum_{i=1}^{n-1} E(f_i(u, v_i)) = (n-1)\beta e^{-\frac{1}{\alpha}}$$

where  $n$ =total number of nodes. Obviously,  $\lim_{n \rightarrow \infty} (E(\mu)) = \infty$ .

We can further prove the divergence of the Waxman’s model.

Let’s consider the average degree of network from the overall point of view.

Let  $E(n)$  be an undirected graph generated by  $n$  nodes, then it is necessary at least to try  $N \geq \frac{n(n-1)}{2}$  times connection operations in order to guarantee the connectedness of graph and equal connection probability for all of the nodes.

For the whole graph, we have average degree of graph  $\mu = \frac{2 \times E_n}{n}$ , where  $E_n$  denotes the total number of links.

According to the above-mentioned analysis, we have

$$E_n = \sum_{i=1}^{n-1} \sum_{j=1}^{n-i} f(u_i, v_j) \geq \frac{n(n-1)}{2} \beta e^{-\frac{1}{\alpha}}$$

then  $\bar{\mu} = E(\mu) \geq (n-1) \beta e^{-\frac{1}{\alpha}}$  where  $\mu = \frac{2 \times E_n}{n}$ .

From above, the network average degree driven from Waxman model increases rapidly with the increasing scale of stochastic network. It suggests the model divergence itself.

After extensive simulation of its performance, we can find out some defects. As for networks with the same scale,  $\alpha$  and  $\beta$ , the average degree resulting from simulation would always differ much. And if the network scale changes, the average degree would not have any consistency. Consequently, some distinct improved models were brought out on basis of RG1 in order to improve such defects, such as:

Exponential Model<sup>[3]</sup>: relate the distance between nodes to probability  $p(u,v)$ , and then the probability of total edges generated would present a downward trend of exponential as the distance between nodes increases. The probability formula is as

following:  $p(u, v) = \beta e^{\frac{-d(u,v)}{L-d(u,v)}}$

Doar-leslie Model<sup>[4]</sup>: factor  $\mu k/n$  (where  $\mu$  is estimated average degree,  $n$  is number of nodes in the network, and  $k$  is a constant) is used to adjust probability  $p(u,v)$  to control the number of edges generated in the network. The probability formula is as

following:  $p(u, v) = \frac{\mu k}{n} \beta e^{-\frac{d(u,v)}{\alpha L}}$ .

Further research discovered that the convergence of average degree in the models described above is still not satisfied. There are still differences between the situation of real network and the ratio of long edge to short edge as well as the probability of connection of over-lengthy edges. The high connection rate of core nodes and the defect of impact on network routing of special node clusters (such as overlay functional nodes) in the network are neglected. Therefore it becomes a hot issue concerning how to generate a model of overlay network simulation more consistent to the real network in order to support more complicated routing strategies (such as active overlay network routing or partial overlay multicast network).

### 3 EAD Model for Stochastic Network Simulation

A computer network can be abstracted as a graph  $G=(V, E, C)$ , where  $V$  represents the set of nodes in  $G$ ,  $E$  the set of direct edges in  $G$ , and  $C$  the set of weights



corresponding to directed edges. The weight of edge is also called the cost of edge. Every edge corresponds to two nodes  $u, v \in V$ . Every two nodes correspond to two edges  $(u, v)$  and  $(v, u)$ . If the costs of these two edges are equal, the graph  $G$  is called symmetric graph, otherwise, the graph  $G$  is called asymmetric graph, i.e., direct graph. In this paper,  $V$  represents the set of routers in the network,  $E$  represents the set of all links, and  $C$  represents the set of bandwidth, delay, data loss rate or aggregation of them. The research on algorithm to generate graph models for network simulation is based on undirected graphs.

### 3.1 Generation of EAD Stochastic Nodes

The nodes in stochastic network mentioned above are all scattered in a fixed plane according to probability of uniform distribution. What is worth of attention is that, nodes in real network are not distributed uniformly, but often incline to partial center-focused distribution<sup>[5]</sup>. For example, in domestic network, the distribution density of network nodes in the east inshore area with dense population and more developed economy is generally heavier than that of west of China with sparse population and less developed economy. As for international network, this trend is more obvious. The distribution density in developed countries is heavier than that in developing countries. In addition, there is hardly a network node in oceans, which covers more than 75% of the surface of the Earth. In order to eliminate this difference, EAD makes some improvements to distribution of network nodes as follows:

The whole network is divided into several sub-areas, which is marked as dense areas ( $D$  areas), sparse areas ( $S$  areas) or 0 distributing areas ( $Z$  areas) according to actual requests (or randomly). When generating a network, nodes can be generated with different densities.

In real network (e.g., the Internet), the degree for most nodes is between 3 and 4, and only a few nodes at network center (3% up to 2000) could exceed 20. These core nodes (expressed with set  $O$ ) include ISP suppliers, large-scale research institutions, industrial network centers, telecommunication providers and so on. Previous simulation models did not consider the degree of core nodes; therefore the constraints of all nodes are all the same without distinguishing the primary and the secondary. There is a great disparity with real network, so it will unavoidably bring uncertain factors to the network simulation of some algorithms.<sup>[6]</sup>

Furthermore, traditional models for stochastic network simulation did not involve the concept of special functional node cluster in network (such as overlay nodes, and multicast nodes). But these nodes may be supporting a certain service of the whole network. For example, there are nearly 20% of the nodes in real network supporting multicast function (17% in 2000). These nodes are able to execute some applications throughout the network<sup>[7]</sup>. How to generate a network model containing special functional nodes, and how to offer the simulation verification for related applications become a very realistic problem.

### 3.2 Generation of EAD Stochastic Linkage

According to the previous explanation, if  $\alpha$  and  $\beta$  do not change, the average node degree will increase along with the increase of the network scale  $n$ <sup>[8]</sup>. In order to overcome such impacts, formula (4) has to be modified. As we know, when

$p=O(\log_{10}n/n)$  ( $0<p<1$ ), the bound of maximum degree and minimum degree of stochastic network are already determined. Therefore the average degree of network will not increase unlimitedly. So, we introduce the following formula through comparison of repeated convergence and experimental verification:

$$p(u, v) = \frac{F_1 \log_{10} n}{n} \beta e^{\frac{-d(u,v) \ln n}{L\alpha F_2}} \tag{4}$$

We can deduce from formula (4) that connection probability  $p(u,v)$  approaches to 0 as network scale  $n$  increases. In this way, the average degree of network adopting formula (4) will not increase with the increase of network scale  $n$ . Our experiments also discover that when  $F_1=16$ ,  $F_2=2.8$ , the average degree will have a very good convergence. At the same time, if  $F_1$  increases, network connectivity will increase; if  $F_2$  decreases, the scale of long edge will decrease further. So it is suitable to apply formula (4) in the case of connection with non-core nodes. But, formula (4) also brings some problems. According to formula (4), the scale of long edges will tend to decrease with the increase of network scale  $n$ . That means all nodes tend to connect to adjacent nodes. It is obviously unsuitable for some long distance connections between core nodes and some special nodes among regions and countries. So we introduce formula (4) as a supplementary restriction such as

$$p(u, v) = \beta e^{\frac{-d(u,v)}{\alpha L}} \tag{5}$$

In order to make the average degree of network more exact, EAD makes further restraint to the generation of stochastic network:

When there is no any core node between two connected nodes, the formula of connection probability (4) is adopted, at the same time some restrictions are added: if the degree of any node of the two exceeds  $\mu+1$ , then the other nodes can not connect to it, where  $\mu$  is the expected average degree. However, if one of the two nodes is the core node, then the formula (5) is adopted, where the upper bound of core node is 12 or 25 (when  $n>64$ ), i.e., when the degree of the core node is bigger than 12 or 25 (when  $n>64$ ), EAD will not permit other nodes to connect to it.

In the process of connection, in order to keep the consistency of connection probability for each node, we adopt the algorithm of connecting out-nodes in order, while select unmarked in-nodes (nodes with marks will not participate in computation) randomly. We select an in-node for current computation of connection probability, and reject the connection generated by the same two nodes repeatedly. In course of computation, when the degree of an out-node equals to  $\mu+1$  (non-core node), 12 or 25 (core node and  $n>64$ ), the computation of this node is dropped, and this node should be marked (as not to participate in computation again). Then the next out-node is orderly selected to continue the computation. As for in-node, if it has already satisfied the condition of dropping computation, the only thing we should do is just to mark it (as not to participate in computation), and then select the next node randomly to continue.

Experiments show that EAD model for stochastic network generation accords with existing real network both in average degree and in the ratio of long edge to short edge better than traditional models.

### 3.3 Fast Connection Strategy of EAD

In the process of generating connections, the existence of every connection is independent to one another; therefore the resulting stochastic network may not be in connectivity for all of nodes. The method in traditional models is to make test at the end of the generation of connections of nodes. If this network is not a singly connected one, we usually reject it and re-generate another one until we get a singly connected stochastic network.

Experiments show that we have to attempt many times to obtain a singly connected network, when there is a large number of a node in the network. There is nearly an exponential relationship between the average attempt times and connection rate (the rate of total connected nodes to total nodes). For example, if 200 nodes are connected completely and stochastically, the required average attempt times would be more than 5000, but if only 95% of nodes need to be connected, the required average attempt times would be no more than 20. This is of much significance to obtain a single-connected stochastic network for a large scale network. EAD will first apply the generation method mentioned above to get a quasi-connection network with more than 95% nodes connected.

The other unconnected nodes or subnets could be divided into two categories: (1) absolutely isolated nodes; (2) some small subnets in which the nodes are connected completely. For (1), each absolutely isolated node executes the Dijkstra algorithm<sup>[9]</sup>, and makes itself linked to the quasi-connection network through the shortest path; For (2), every node in each small subnet executes the Dijkstra algorithm, and finally the subnet will seek a shortest path to connect to the quasi-connection network. Thus, EAD can spend much less time to get a completely connected stochastic network.

## 4 The Testify of Convergence of EAD

For the node connectivity probability formula (4), since,  $0 < d(u,v) < L$ ,  $\alpha, \beta, F_1$  and  $F_2$  are constants, obviously, we have

$$\lim_{n \rightarrow \infty} P_n(u, v) = \lim_{n \rightarrow \infty} \frac{F_1 \beta \log_{10} n}{n} \cdot e^{\frac{-d(u,v) \ln n}{L \alpha F_2}} = 0 .$$

That is, the node connectivity probability approaches zero as a limit with the increasing of network scale  $n$ .

If  $p$  is the connection probability of a node with other nodes in a stochastic network graph, which contains  $n$  nodes, its average degree probability distribution is

$$P(\mu = k) = C_{n-1}^k p^k (1-p)^{n-1-k} \tag{6}$$

On the basis of probability theory, when  $p < 1$  and  $n$  is large enough, binomial distribution (6) can be expressed by Poisson distribution. If choosing  $\frac{(n-1) \times n}{2} \times p$  edges with equal probability to construct a graph, the probability distribution of the times of each node chosen is the average degree of this graph. If the chosen probability of a node is  $\rho$  in the process of connection, then each node has  $n-1$  connections that can be chosen at each time when connection is chosen. Because of the equality of choosing probability and each connection with two nodes, we have  $\rho = 2/n$ . When  $p < 1$ , the choosing of any node does not affect the re-chosen probability of another node connecting with it. So, choose the node with probability  $\rho$  composes probability space, which fitted the Poisson distribution and its average value is  $\lambda = \rho \times \frac{(n-1) \times n}{2} \times p = (n-1)p < np$ .

That is  $p(\mu = k) = \lambda^k e^{-\lambda} / k! = (np)^k e^{-\lambda} / k!$ . In an  $n$ -scale stochastic network, the connectivity of the node  $u$  is  $\mu = \sum_{i=1}^{n-1} f(u, v_i)$ . Where,  $f(u, v)$

$$f(u, v) = \begin{cases} 1 & p(u, v) \geq \text{rand}(0, 1) \\ 0 & p(u, v) < \text{rand}(0, 1) \end{cases}$$

For the degree  $u$  of single node, we have  $E(\mu) = E(\sum_{i=1}^{n-1} f(u, v_i)) = \sum_{i=1}^{n-1} E(f(u, v_i))$ .

To the  $\text{rand}(0, 1)$ ,  $y$  is uniform distribution stochastic variable and follows the distribution:  $f(y) = \begin{cases} 1 & 0 < y < 1 \\ 0 & \text{else} \end{cases}$ .

If stochastic variable  $x = p(u, v)$ ,  $d(u, v)$  is a variable. And  $d(u, v)$  follows the uniform distribution in  $(0, L)$ . Let  $d(u, v)$  be stochastic variable  $D$ , then it follows

$$f(d) = \begin{cases} \frac{1}{L} & 0 < D \leq L \\ 0 & \text{else} \end{cases}$$

where,  $D$  is the stochastic variable.

Then  $x$  can be expressed as the function of  $D$ :  $x = \frac{F_1 \beta \log_{10} n}{n} \cdot e^{\frac{-D \ln n}{\alpha F_2}}$

$$\begin{cases} f(x) = \frac{F_1 \beta}{x \ln n} & (\frac{F_1 \beta \log_{10} n}{n} e^{\frac{-L \ln n}{\alpha F_2}} < x < \frac{F_1 \beta \log_{10} n}{n}) \\ f(y) = 1 & (0 < y < 1) \end{cases}$$

$$f(u, v) = P\{x > y\} = \iint_{x>y} f^*(x, y) dx dy$$

Obviously,  $x$  and  $y$  are independent to each other.

$$f(u, v) = P\{x - y > 0\} = \iint_{x>y} f(x) f(y) dx dy$$

The following can be derived easily

$$f(u, v) = \frac{F_1 \beta}{n} \left( \ln \frac{\beta F_1 \log_{10} n}{n} - \frac{\log_{10} n}{n} \beta F_1 \cdot e^{\frac{-\ln n}{\alpha F_2}} \right)$$

$$\bar{\mu} = E(\mu) = (n - 1) f(u, v)$$

When  $n \rightarrow \infty$ , then  $E(\mu) \rightarrow F_1 \beta$ . That is when  $n \rightarrow \infty$ , the main connection strategy of EAD model is convergent, and converges to  $F_1 \beta$ .

### 5 Performance Comparison of EAD with Traditional Models

When comparing our model with traditional models, we set out from the following principle: under the condition of equal average degree, we select parameters more close to actual network characteristics<sup>[10]</sup>. That is to regard characteristics most close to the real network as precondition to adjust according to the different adjusting function to connection probability formula of  $\alpha$  and  $\beta$ . Refer to Tables 1 and 2 for the comparison of average degrees of RG1 and EAD.

According to the comparison of characteristics of node degrees of models above, we can find out that EAD is obviously better than RG1 in the aspect of average degree. When network scale rises to 1024, RG1 average degree has already arrived at 89.17. In fact, since such stochastic network generated shows much difference to real network, so there is no realistic meaning. However, when we control  $\alpha=0.01$  or so, for the stochastic network R generated by G1, although the average degree can still bear preferable convergence if network scale is larger than 512, it does not accord with real network seriously due to over-high loss rate of long side. In fact, this network does not bear much realistic meaning. As for the characteristics of average degree, RG1 and exponential model are fundamentally the same: when network scale increases continuously, the algorithms present strong ascending tendency of average degree with very poor convergence. Although exponential model is better, it cannot adjust  $\alpha$ . Because  $\alpha$  is responsible for controlling the ratio of long side to short side, there is a great gap between the resulting simulation network and the real network.

As seen from those Tables, after EAD algorithm applies restraint in node degree, its convergence becomes excellent, and the average degree is always limited within a very small range. At the same time network scale expands with multiples. Because when the distance between two nodes increases, EDA algorithm will apply formula (4) to calculate the connection probability. Actually connection probability will decrease with exponent as network scale increases. However, many long edges connected to core nodes in real network will be lost while EDA algorithm guarantees the average degree. As a result, EAD algorithm also takes into account long-distance

**Table 1.** Comparison of RG1 and EAD Average Degree Where:  $F_1=16, F_2=2.8, L=100\sqrt{2}$

Model	Scale of network	16	32	64	128	256	512	1024
RG1	$\alpha = 0.2, \beta = 0.7$	3.78	6.37	9.70	19.26	37.21	68.31	89.17
	$\alpha = 0.01, \beta = 0.9$	2.19	2.42	2.96	3.27	4.56	6.14	8.19
ONSM	$\alpha = 0.4, \beta = 0.6$	$\mu = \infty$	5.42	5.23	5.32	5.47	5.65	5.49
		$\mu = 4$	4.23	4.17	4.26	4.21	4.16	4.11

**Table 2.** Comparison of Various Models

Model	Scale of network	16	32	64	128	256	512	
RG1	$\alpha=0.2, \beta=0.7$	3.92	6.41	10.26	23.82	46.79	56.11	
Exponential	$\beta = 0.25$	4.12	4.85	5.16	6.03	6.90	7.83	
Doar-Leslie	$\alpha=0.15, \beta=0.3, k=20, \mu=4$	4.01	7.28	9.59	18.72	32.90	53.64	
ONSM	$F_1=16, F_2=2.8 \alpha=0.4, \beta=0.6$	$\mu = \infty$	5.42	5.13	5.32	5.47	5.65	5.49
		$\mu =4$	4.23	4.17	4.26	4.21	4.16	4.11

connection of core nodes to other nodes while it adjusts the ratio of long edge to short edge through  $\alpha$  and  $\beta$ . By adopting formula (1), and calculating connection probability between a core node and another node, EAD algorithm prompts to compensate the lost long edge while guarantee connection of high node degree in core nodes 12; 25 ( $n>64$ ), so that the generated simulation network is much closer to real network than before.

EAD algorithm shows significant improvements over the traditional models. No matter in the aspect of core node and its behaviors of high connection node degree, or in the aspect of embodiment of special function node in the network, or in the aspect of convergence of average degree that we have been caring in the past, EAD algorithm is closer to real network. EAD algorithm provides a good experimental platform for the universal network simulation based on the stochastic network model.

## 6 Conclusion

This paper studies the impacts on simulation network of core nodes, node clusters of special functions, convergence of average degree and ratio of long edge to short edge. We propose a method to generate simulation network from a new aspect. The simulation network generated by EAD algorithm is further approaches to real network. The EAD method provides an excellent experimental platform for network simulation of related applications, especially simulation of special networks (such as active overlay network or partial multicast network). if we assign a value from 0 to Max randomly while EAD generates real connection between two nodes, we can simulate to generate an asymmetric network. And if we execute multi-target limites to special functional nodes in the algorithm, we can provide support to the simulation of various overlay networks.

There is still a lot of work to do on the algorithm research of stochastic network model. Aiming at different demands (such as new-type polyhedron layered routing strategy), we can make necessary alteration to EAD. However, as for exact average degree, it has already laid a good foundation for further research in the future.

## References

1. Dong, Q.: Research on Computer Communication Network Multicast Routing Algorithm and Protocol. PhD Thesis, Shanghai Jiaotong University, Shanghai, China, (2000)
2. Waxman, B.M.: Routing of Multipoint Connections. IEEE Journal on Selected Areas in Communications, 6(9) (1988) 1617-1622

3. Berry, L.: Graph theoretic models for multicast communications. *Computer Networks and ISDN Systems*, 20(1)(1990) 95-99
4. Doar, M.B.: A Better Model for Generating Test Networks. In *Proceedings of Globecom '96. Global Internet '96* (1996)
5. Wang, Z.: Research on High-speed Network QOS Routing Technology and Development of Routing Simulation Platform. PhD Thesis, Huazhong University of Science and Technology (2001)
6. Li, H.: Computer Network Routing Algorithm. Doctoral academic dissertation, Xidian University (2000)
7. Wang, X.: Development of System Simulation Technology in the 21st Century. *Journal of System Simulation*, 11(2) (1999)
8. Bajaj, S., Breslau, L., Estrin, D., Fall, K., etc. : Improving Simulation for Network Research, Technical Report 99-702, University of Southern California, (1999)
9. Dijkstra, E.W.: A note on two problems in connection with graph. *Numerische Mathematik*, 1(3) (1959) 269-271
10. Olabe, M.A., Olabe, J.C.: Telecommunication Network Design Using Modeling and Simulation. *IEEE Transaction on Education*, 41(1) (1998) 37-44

# Applying Semiotic Analysis to the Design and Modeling of Distributed Multimedia Systems

Mangtang Chan<sup>1</sup> and Kecheng Liu<sup>2</sup>

<sup>1</sup> Department of Computer Science, City University of Hong Kong,  
83 Tat Chee Avenue, Kowloon, Hong Kong SAR, PRC  
csmtchan@cityu.edu.hk

<sup>2</sup> Department of Computer Science, School of Systems Engineering, University of Reading,  
Whiteknights, Reading, Berkshire, RG6 6AY, United Kingdom  
k.liu@reading.ac.uk

**Abstract.** Semiotics is the study of signs. Application of semiotics in information systems design is based on the notion that information systems are organizations within which agents deploy signs in the form of actions according to a set of norms. An analysis of the relationships among the agents, their actions and the norms would give a better specification of the system. Distributed multimedia systems (DMMS) could be viewed as a system consisted of many dynamic, self-controlled normative agents engaging in complex interaction and processing of multimedia information. This paper reports the work of applying the semiotic approach to the design and modeling of DMMS, with emphasis on using semantic analysis under the semiotic framework. A semantic model of DMMS describing various components and their ontological dependencies is presented, which then serves as a design model and implemented in a semantic database. Benefits of using the semantic database are discussed with reference to various design scenarios.

## 1 Introduction

With the advancement of the World Wide Web and the tremendous hardware capability of desk-top and server machines, nearly all recent applications would involve the use of multimedia. In the design arena, however, DMMS are treated just as normal software systems and traditional methodologies are used. The prevailing one is the Object Oriented design using UML [1] as the design language. A closer look at the design issues of DMMS would raise questions that are not found in traditional software system design. DMMS involve diverse data types with much more complex structures compared to traditional text and numbers; data volume is huge in terms of storage and transmission, and data presentation is time dependent. Good design for DMMS depends very much on the experience and knowledge of the system designers. To solve this problem, design knowledge is shared by documenting design constructs as design patterns [2]. Design patterns are grouped by application domains. Use of them may have a number of limitations. Firstly, the learning curve could be steep and secondly, the knowledge of why the pattern is designed in such a way is not captured. Finally, it is difficult to implement code generation without a knowledge base of the application domain. Design pattern supports reuse of design knowledge from previous



designs but it does not support cooperative collaboration among designers working at different parts of a system at the same time. The use of CASE tools could, to some extent, help in collaboration but CASE tools that could support real-time sharing of design information with good compatibility are not commonly known. In collaborative design, three aspects are identified as important: availability of knowledge from previous design; immediate sharing of design information among co-designers, and ease of reuse by importing and exporting.

Existing methodologies do not provide a well integrated design environment for DMMS. Agent-based software development methodologies [3], on the other hand, offer a promising software engineering approach for developing applications in complex domains. There are a number of methodologies in agent-based software systems design and they have been recently applied to the design of multimedia systems.

Regarding an information system as a collection of agents interacting with each other is one of the main theme study of Semiotics – the study of signs [4]. Philosophically, semiotic study proposes the functioning of an information system is the result of active interpretation of the environment by agents. Technically, semiotics develops into a set of tools for analysis and design of information systems [5], namely, the Problem Articulation Method (PAM), Semantic Analysis Method (SAM) and the Norm Analysis Method (NAM). Semiotic analysis has recently been applied in the high level study of organizations. The semiotic framework emphasizes the use of semantic analysis, supported by a semantic database. It is therefore a knowledge based methodology. It not only enhances collaboration by sharing knowledge but at the same time could capture more application domain knowledge through interaction with designers. The accumulated knowledge would be particularly valuable to current and subsequent designers in dealing with complex systems such as DMMS.

The contribution of this paper is the application of semantic analysis under the semiotics framework to build a knowledge base for DMMS design. The remaining parts of the paper are organized as follow:-

- review of related work to provide the background for the design approach proposed by the authors
- discussion of the semantic analysis of DMMS
- presentation of a semantic database for capturing semantic information and design knowledge
- analysis of some application scenarios of the semantic database in the context of collaborative design
- concluding remarks and future work

## 2 Related Work

Starting from the early 90s, various approaches had been used in DMMS design and modeling. In the system architecture area, Baker et al [6] used a layered model with abstractions such as stream, multimedia presentation and hyper-presentation to describe distributed multimedia I/O systems. Lots of work had been done in building workstations, storage systems and network that were suitable to work with multimedia, e.g., Shenoy, Goyal and Vin [7], Vin, Goyal and Goyal [8] and Stallings, [9]. In the performance area, Blair et al [10] compared a number of formal languages and

selected LOTOS and temporal logic to model the QoS properties of multimedia systems. Challenges in the demand of QoS and synchronization were also well studied, e.g. Nahrstedt, [11] and Gaggi [12]. In the design and implementation areas, Posnak [13] adopted an object-oriented framework to develop a Presentation Processing Engine (PPE) to allow easy code reuse. Applications could be built by linking the objects through a scripting language, Tcl/Tk [14]. In summary, designing DMMS has to address three different but closely linked aspects: structural, spatial and temporal. Chan [15] proposed a set of requirements for an integrated design environment for DMMS and asserted that they have not been fulfilled by existing design models and methodologies.

In semiotics, Stamper [4] laid down the fundamental work of sign, information, norm and system. He proposed a six layers semiotic framework that can be used to analyze information systems. The layers are:

- Physical World – dealing with signals, physical hardware, speed, etc.
- Empirics – dealing with pattern, noise, codes, channel capacity, etc.
- Syntactics – dealing formal structure, language, logic, software, files, etc.
- Semantics – dealing with meaning, propositions, validity, truth and denotations etc.
- Pragmatics – dealing with intentions, communications, conversations and negotiations, etc.
- Social World – dealing with beliefs, expectations, functions, contracts, culture and law, etc.

The first three layers form the part that is usually referred to as IT platform while the last three layers involve human information functions. Stamper also proposed the notion that information systems are organizations within which agents exhibit actions or signs according to a set of rules (norms). An analysis of the agents, their capabilities (affordance) and the norms would give a better specification of the information than the traditional data flow model. This formed the basis for semantic analysis in the semiotic framework. Liu [5] further summarized the semiotic semantic analysis into four major phases: Problem definition; Candidate affordance generation; Candidate grouping and Ontology charting.

The candidates are semantic units or vocabularies used in the semantic model being analyzed. They can be further classified into different types, e.g. agents, affordance or role. Ontology charting is the analysis of relationships between different agents and affordances. Semantic analysis and ontological dependency under the Semiotic Framework had been used to design different business information systems for project management, land resources management and examination test construction. All these work demonstrated ontological analysis and knowledge-driven agent based systems can be an alternate methodology to the prevailing UML based object-oriented design or the more traditional Structured System Analysis and Design Methodology (SSADM) for information systems design.

In semiotics, there is no definition about the nature of the agents. They could be human, software or even a department within an organization. The characterization, however, shows remarkable similarity to agents as defined in the research community of intelligent agents and multi-agent systems. Wooldridge [3] in his survey of agent-oriented software engineering methodologies pointed out that agent-oriented software

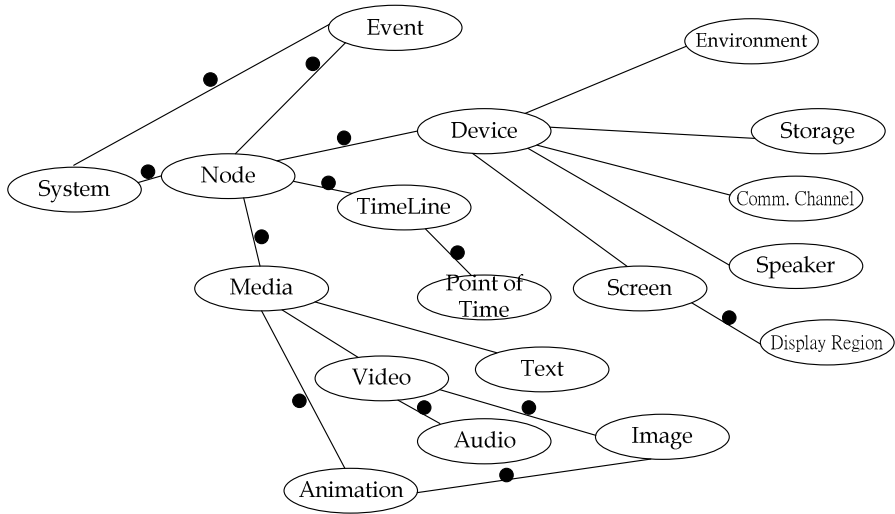
engineering is still at an early stage of evolution. There is no discussion in the original semiotic framework about implementation. The original work done by Liu [5] involved the use of the LEGOL language and object-oriented programming environment to implement the semantic model. More recently, Filipe [16] used the semiotic approach to specify and engineer agent-oriented organizational information systems. This has shown that semiotic framework could also serve as another methodology for designing agent-oriented software systems. DMMS, architecturally match very well with multi-agent systems. The authors therefore take the view that applying semantic analysis to the design of DMMS would be a research opportunity. We will also show later in this paper the benefits of using this approach for collaborative design.

### 3 Semantic Analysis for DMMS

The authors propose to apply semantic analysis techniques as defined under the Semiotic Framework [5] to design DMMS. Semantic analysis is the process to study the relationships between agents and affordance, and agents and agents. The end result of this semantic analysis is not a particular design for a certain DMMS, but rather the semantic knowledge in the domain of DMMS design which would be useful in later design of specific application systems. Since the scope of this paper covers only part of the analysis and design cycle, other steps in the semiotic framework such as norm analysis will not be discussed.

As a start, candidates or semantic units in the domain of DMMS would be identified. Building an ontology for DMMS is very useful in this aspect. Ontology has been used in the Artificial Intelligence (AI) community and recently in software engineering. Chandrasekaran [17] looked at the application of ontologies in AI and information systems. Devedzic [18] showed the analogies between ontological engineering and software engineering. It should be noted that ontological dependency analysis in semiotics is different from ontologies used in AI community, although some similarities exist. This paper extends the original semantic analysis used by Liu [5] to incorporate the building of an ontology for DMMS before the ontological dependency charting. The ontology mainly deals with the taxonomy, the generic-specialized and compositional relationships of semantic units which are the candidates to be analyzed for their ontological dependency.

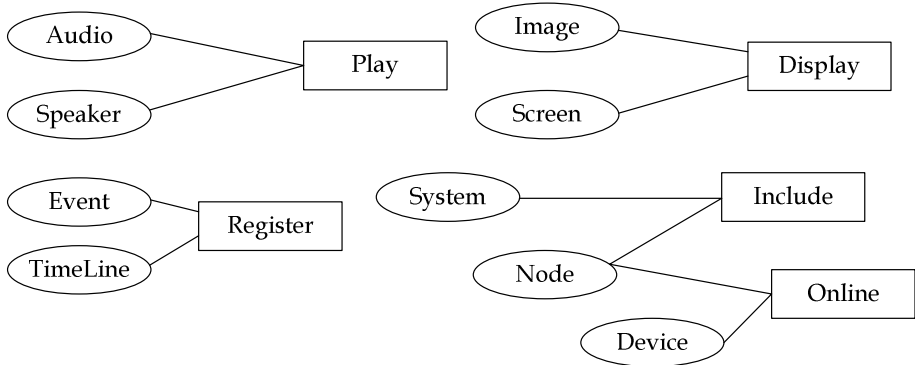
Two types of charts are used, one for describing the overall taxonomy of the system and other for ontological dependency analysis. To avoid confusion, the former would be referred to as ontology chart while the latter referred to as ontological chart. Semantic units in the DMMS are collected to form a taxonomy tree using two relations, “part of” and “type of”. The four notions proposed by Guarino [19], identity, rigidity, unity and dependency were used to validate and produce the so-called “backbone taxonomy”. The semantic units chosen are likely nouns or objects that would be involved in some operations to achieve some purposes in the DMMS domain. These would help to identify the agents and affordances. Figure 1 depicts part of the ontology for DMMS where an eclipse symbol stands for agent, a line represents the relation “is a type of” and a line with a dot represents the relation “is a part of”. Figures 2, 3 and 4 are ontological charts showing relationships among Agent, Affordance, and Role.



**Fig. 1.** Part of the ontology for DMMS

In the ontology chart, a system consists of one or more nodes, which are interpreted as computer systems at a certain location. If the system has one node, it is a multimedia system involving one computer e.g. a desk top computer or a DVD player. A number of nodes connected together in different locations would then become a DMMS. A node consists of one or more devices. Speaker and screen are device types. A node also consists of, or more precisely, supports the operations of some media types, which could be video, audio or text, etc.

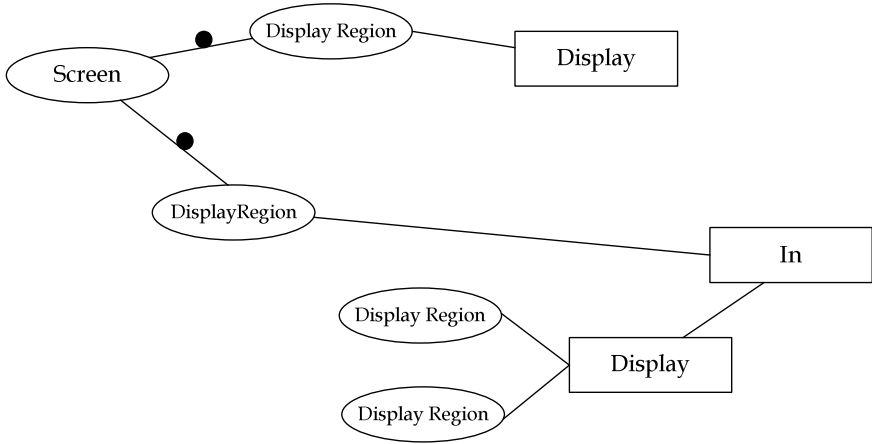
After the ontology is drafted, the ontological charts will be completed by firstly, finding out the affordances of the agents and secondly, enumerating the relationships



**Fig. 2.** Connection of agents through affordance

between each pair of agents and checking if they could be linked by one or more of the affordance. Some examples are: a node can connect to a system; a device can be online (register) with a node; or an image can be displayed itself on a screen and etc. Figure 2 contains some of these relations. The ontological charts carry very rich semantic information about DMMS. Some more details examples are discussed in the following figures.

The layout of a web page can be modeled by the ontological chart shown in Figure 3. A web page can be displayed on a screen containing two display regions with one further divided into two regions, one for text display and the other for image. This indicates the Display Region could be recursively defined.



**Fig. 3.** The ontological chart of part of a Web page

Modeling of streaming video is interesting in the sense it uses the concept of Role in a relation. In Figure 4, a video is “streamed” to a TV with stream, as an affordance, connecting the two agents, Video and TV. The video participates in this connection with a role of source. On the other hand, if a Camera participates as source and the video as destination, then capturing of a live video feed scenario will be modeled. In actual modeling, instances of a device will be specified and it is obvious that a speaker could not participate in a stream with a role of source, but a microphone could. Video instances of format MPEG and QuickTime can be connected to agent Codec (coder and decoder). Their roles as either Source or Destination would be defined by the capability of the Codec concerned. The more general type of agent Device therefore should not be used in this case otherwise the semantics defined will be that all devices could connect to a stream as a source. This implies the semantic link between agent-affordance-agent is directional (right dependent on left) and the existence of the link carries the information about the authority of the preceding agent for the affordance. This illustrates some of the basis for semantic checking during design modeling.

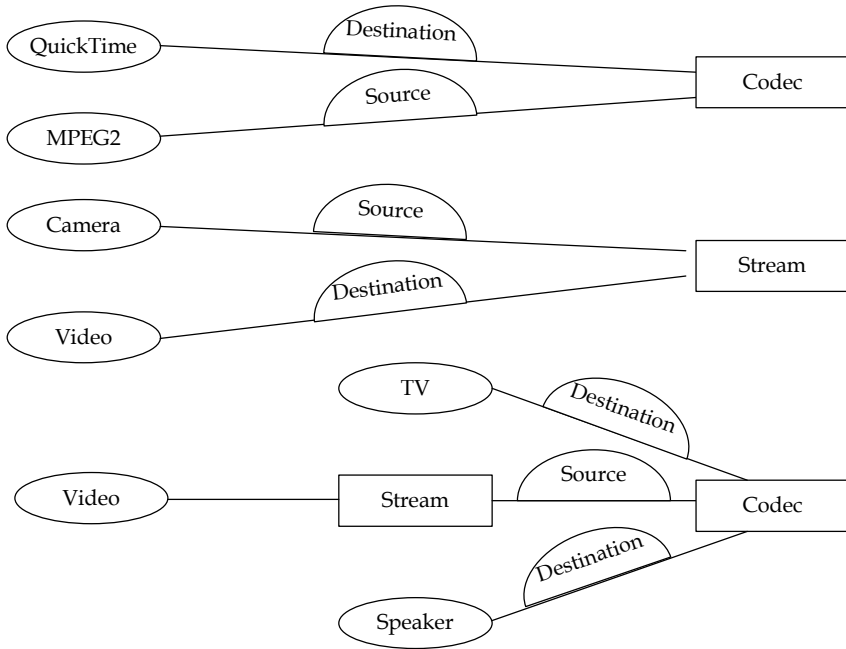


Fig. 4. Use of the role concept in ontological chart

#### 4 Dynamic Representation of the Semantic Information

Ontological chart is suitable for presenting a visual description to users. For subsequent manipulation, it has to be represented, preferably in a formal way. This paper uses a modified version of semantic database (SDB) proposed by Liu [5] for semantic information representation. It is conceptually more generic and not difficult to be implemented with programming tools. The SDB stores the ontological charts in the form of tuples with only one structure :

*<id, semantic unit, sort, semantic unit type, antecedent(s), link type>*

Some sample entries in the database used to define parts of preceding figures are :

- <1, System, type, agent, -, -, ->*
- <2, Node, type, agent, System, -, part-of>*
- <3, Media, type, agent, Node, -, part-of>*
- <4, Video, type, agent, Media, -, type-of>*
- <5, Audio, type, agent, Video, -, part-of>*
- <6, Image, type, agent, Video, -, part-of>*
- <7, Stream, type, affordance, Video, TV, ->*
- <8, Stream, type, affordance, Camera, Video, ->*
- <9, DisplayRegion, type, agent, Screen, -, part-of>*

<10, *DisplayRegion*, type, agent, *DisplayRegion*, -, part-of>  
 <11, *Source*, type, role, *Video*, *Stream*, -, ->  
 <12, *Destination*, type, role, *TV*, *Stream*, -, ->

The “-“ character stands for a null entry. The *id* serves as a global identifier and would be useful in referring to any entry in the SDB. The *sort* defines whether the entry is a type definition or an instantiation of a type definition. There could be none, one or two antecedents depending on the relationship of current entry with other entries. The *link* type is to distinguish whether the relationship of two agents, is a “type of” or “part of” relation. The former defines a generalization-specialization relationship to support the concept of inheritance and the latter represents the part-whole relationship.

## 5 Use of the SDB in DMMS Design

Designers would interact with the SDB to build models of the system to be designed. If a desk top computer with a web camera and a microphone, able to play Quicktime movie, is to be modeled, the following entries could be inserted into the SDB:

<13, *mySystem*, instance, *System*, -, -, ->  
 <14, *myDesktop*, instance, *Node*, *mySystem*, -, part-of>  
 <15, *myDev*, instance, *Device*, *myDesktop*, -, part-of>  
 <16, *myMed*, instance, *Media*, *myDesktop*, -, part-of>  
 <17, *Camera*, type, *Device*, -, -, type-of>  
 <18, *wCam*, instance, *Camera*, *myDev*, -, type-of>  
 <19, *myMic*, instance, *Microphone*, *myDev*, -, type-of>  
 <20, *Quicktime*, type, *Video*, -, -, type-of>  
 <21, *myVideo*, instance, *Video*, *myMedia*, -, type-of>  
 <22, *myMov*, instance, *Quicktime*, *myVideo*, -, type-of>

The following entries could be used to specify a video-on-demand (VOD) system. A VOD system has the characteristic of run time connection of video streams.

<23, *vodSystem*, instance, *System*, -, -, ->  
 <24, *node1*, instance, *Node*, *vodSystem*, -, part-of>  
 <25, *node2*, instance, *Node*, *vodSystem*, -, part-of>  
 <26, *screen1*, instance, *Screen*, *device1*, -, type-of>  
 <27, *screen2*, instance, *Screen*, *device2*, -, type-of>  
 <28, *server*, instance, *Node*, *vodSystem*, -, part-of>  
 <29, *video1*, instance, *Video*, *serverMedia*, -, type-of>  
 <30, *strm1*, instance, *Stream*, *video1*, *screen1*, ->  
 <31, *source1*, instance, *Source*, *video1*, *strm1*, -, ->  
 <32, *des1*, instance, *Destination*, *screen1*, *strm1*, -, ->  
 <33, *stream2*, instance, *Stream*, *video1*, *screen2*, ->

<34, *source2*, *instance*, *Source*, *video*, *strm2*, -, ->

<35, *des2*, *instance*, *Destination*, *screen2*, *strm2*, -, ->

For clarity, some entries are left out or could be assumed to be present in the SDB. Creation of the entries looks tedious but once a tool or user interface is available, it should not be difficult. It should be noted these entries could be created during design time or run time. Run time creation (i.e. created by an agent instead of the designer) is more suitable for a VOD system because dynamic creation of video streams is usually a requirement. In this example, three nodes are involved with one server serving two nodes. The same video file, i.e. *video1* is acting as the source to both video streams to the two destination screens. The same video file could be using two different time lines for presentation in a point-to-point mode or using one time line in a broadcast mode.

The original version of the SDB of Liu [5] was a temporal database with non-destructive updating. All entries created would stay and the SDB would grow with additional knowledge being captured all the time. To allow for versioning and distinguishing active instances, time attributes would be included in the entry format.

<*id*, *semantic unit*, *sort*, *semantic unit type*, *antecedent(s)*, *link type*, *active time*, *end time*>

The active time marks the time of action defined by the entry. End time indicates the entry has become a historical datum in the SDB. Instances are therefore active if their action times have values and end times are empty. Instances with end time values are no longer available in a running system.

## 6 Use of the SDB in Collaborative Design

In the introduction of this paper, three aspects of collaboration are identified, namely, design assisted by knowledge of the application domain from previous designers, concurrent design by more than one designers working in a project and reuse of design by importing from other sources. The proposed SDB offers various opportunities to enhance collaboration in all these aspects.

The primary advantage is to have a central knowledge-driven database accessible by all designers. Once an entry is created, that piece of knowledge will be known to all subsequent designers. Design agents can be developed to interface with the SDB and guide the designer to create new system models. By making queries to the SDB, model validation can be done by the design agent and reduce, to some extent, the dependency on the experience of designers. The SDB will not only benefit designers during design time, it could also enable system agents to construct other agents based on run time information. Using the VOD example again, a system agent can monitor the VOD system and creates additional server agents as more and more video streams are requested by client nodes. The number of active streams and the knowledge of how to create server agents are available in the SDB.

For concurrent design, every type or instance information is immediately available to co-designers. For designers working at different parts of a system, they could look at the actual design and progress of each other through a user interface or design



agents. To impose proper concurrency control, an author attribute could be introduced into the SDB entries such as,

*<id, semantic unit, sort, semantic unit type, antecedent(s), link type, active time, end time, author>*

While entries could be created or changed by the authoring designer, entries maintained by other designers would then be read-only.

For design reuse, the concept of a view for the SDB could be defined. A view is a selection of related entries in the SDB based on some criteria. It could be entries belonging to an instance of a system or a node; or it could be entries authored by a specific designer. A view based on time is also possible, although its use may not be intuitive. A view of the SDB could also be exported or imported. An imported view could be put to use immediately and would be a convenient way to reuse design from a proven design or a renowned designer.

## 7 Future Work

This paper describes only the first step in applying semantic analysis in semiotics to the collaborative design of complex DMMS, with emphasis on modeling. A prototype video-on-demand server grid system is now being built to validate the proposed approach. In addition to semantic analysis, norms analysis and the full semiotic framework will be investigated for DMMS design. These work would include semantic analysis of the timing and synchronization aspects of different media, how and what properties of semantic units could be used for system implementation; incorporating quality of service (QoS) study of multimedia systems into the ontology and finally, mapping of the design to implementation by knowledge-driven agent systems.

## References

1. Object Management Group.: Unified Modeling Language, <http://www.omg.org/uml>, (2004)
2. Gamma, E., Helm, R., Vlissides, J.J.: Design Pattern: Elements of Reusable Object-Oriented Software. Addison Wesley (1995)
3. Wooldridge, M., Ciancarini, P.: Agent-Oriented Software Engineering: The State of the Art, First Int. Workshop on Agent-Oriented Software Engineering. Springer-Verlag, Berlin, (2000) 1-28
4. Stamper, R.: Signs, Information, Norms and Systems. in Holmqvist, B., Andersen, P.B. (ed.), Semiotics in the Work Place, de Gruyter, Berlin, (1996) 349-399
5. Liu, K.: Semiotic in Information Systems Engineering. Cambridge University Press, Cambridge (2000)
6. Baker, R., Downing, A., Finn, K., Rennison, E., Kim, D.D. and Lim, Y.H.: Multimedia Processing Model for a Distributed Multimedia I/O System. Proceedings of the 3rd International Workshop for Network and Operating System support for Digital Audio and Video, Nov., Springer-Verlag. (1992) 165-175
7. Shenoy, P.J., Goyal, P., Vin, H.M.: Issues in multimedia server design. ACM Computing Survey, Vol. 27 Issue 4 (1995) 636-639
8. Vin, H.M, Goyal, A., Goyal, P.: Algorithms for designing large-scale multimedia servers. Computer Communication. Vol. 18 Issue 3 (1995) 192-203

9. Stallings, W.: Advances in High-speed Networking. *ACM Computing Survey*, Vol. 28. No.1. (1996) 221-223
10. Blair, G., Blair, L., Bowman, H., Chetwynd, A.: Formal support for the specification and construction of distributed multimedia systems (The Tempo Project). Internal Report MPG-93-23, School of Engineering, Computing and Mathematical Sciences, Lancaster University, (1993)
11. Nahrstedt, K.: End-to-end QoS guarantees in networked multimedia systems. *ACM Computing Survey*, Vol. 27 No. 4, (1995) 613-616
12. Gaggi, O., Celentano, A.: Modelling Synchronized Hypermedia Presentations. Technical Report Series in Computer Science, Dipartimento di Informatica, Università Ca' Foscari di Venezia (2002)
13. Posnak, E.J., Lavender, R.G., Vin, H.M.: An Adaptive Framework for Developing Multimedia Software Components. *Communications of the ACM*, Vol 40, No. 10 (1997) 43-47
14. Ousterhout, J.K.: Tcl and the Tk Toolkit. Professional Computing Series, Addison-Wesley, Reading, Mass. (1994)
15. Chan, M.T., Ng, T.S., Yung, N.H.C.: MVM - A Multimedia Virtual Machine for Design Modeling and Performance Simulation. Proceedings of the International Conference on Information Systems, Analysis and Synthesis (ISAS 98), Orlando, USA, Vol. 3 (1998) 9-15
16. Joaquim, F., Liu, K., Sharp, B.: A Semiotic Approach to Organisational Role Modelling for Intelligent Agents. In: Information, organisation and technology: studies in organisational semiotics. (ed) Liu, K., Boston, Mass., Kluwer Academic Publishers (1999)
17. Chandrasekaran, B., Josephson, J.R., Benjamins, V.R.: What are ontologies, and why do we need them? *IEEE Intelligent Systems* Vol. 14, No. 1 (1999) 20-26
18. Devedzic, V.: Understanding Ontology Engineering. *Comm. of the ACM*, April Vol. 45, No. 4 (2002) 136-144
19. Guarino, N., Welty, C.: Towards a methodology for ontology-based model engineering. Proceedings of ECOOP-2000 Workshop on Model Engineering (2000)

# A Rapid Inducing Solid Model Towards Web-Based Interactive Design

Hongming Cai<sup>1,2</sup>, Yuanjun He<sup>2</sup>, and Yong Wu<sup>2</sup>

<sup>1</sup> School of Software, Shanghai Jiaotong University, Shanghai, P.R. China

<sup>2</sup> Computer Science Department, Shanghai Jiaotong University, Shanghai, P.R. China  
hmcai@sjtu.edu.cn

**Abstract.** Web-based interactive design system works ineffectively for mass of data are transferred on narrow bandwidth network. In order to decrease data transferring in CSCW process, a Rapid Inducing Solid Model (RISM) is provided. First, RISM is built to represent the geometric information and operation structure of product. Then based on RISM components which connected to corresponding surface lists, traditional CSG model is extended to construct an operation history tree. Therefore, common operations set and message mechanism could be built to make displaying model work individually to a certain degree. Based on the ACIS geometry engine, a cooperative design system has been implemented for testing. The results show that the model is effective and also provides a new framework for Web applications.

## 1 Introduction

The technology of Computer Supported Cooperative Work (CSCW) gets more and more attentions recently because it is suitable for people to work in an interactive, distributed and cooperative information environment. However, when different terminal users design interactively during Computer Supported Cooperative Design (CSCD) process, it is very difficult to operate product model and see real-time results because there are many data transforming and data displaying occurred. It is an easy thought for people to solve the problem by decreasing data-transferring and accelerating displaying velocity on the network. Therefore, product model is the key factor to transfer design data efficiently.

In the aspect of Web-based solid modeling, VRML (Virtual Reality Modeling Language) is the common used three dimensions object representation. VRML format can be used to display product in client effectively, but it cannot provide high-level geometry modeling functions [1] such as Boolean operation and curve-surface modeling for it only records surface information of product. That is, the model represented in VRML format does not suit to record the design process, and it is difficult to use this model in the process of detail design or manufacturing. Based on VRML, more powerful systems can be implemented, for example, Kiss [2] provided a real-time modeling method supporting both geometry surface modeling and parameter modeling. Hu and Tan [3] combined with JAVA technology to realize most solid modeling functions except union, intersection, etc.

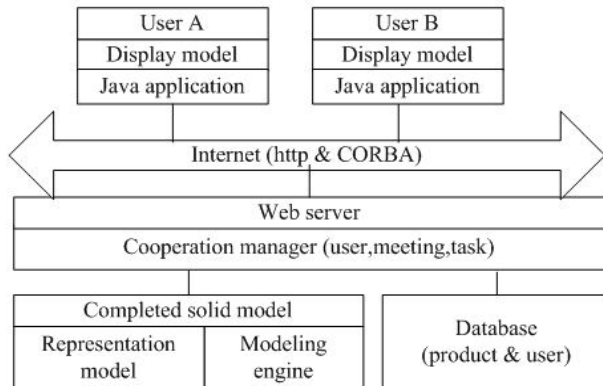
In the aspect of data-transferring on the network, LOD (Level of Detail) technology is the most widely used method. Hoppe [4] proposed the progressive meshes based on edge-collapsed, it could not only present a group of consecutive LOD model, but also support progress transfer pattern. But it fails to reserve important features, errors would be amassed to a big degree. Some methods are not belonged to LOD but also useful for transferring design data economically. Bidarra [5] proposed an idea to carry out design by transferring modified parts only. Zhou et al. [6] proposed an operation driven method to realize synchronized collaborative solid modeling. In our previous work, we proposed an Extend CSG model [7] for recording the operation history information during CSCW process. Zhang et al. [8] presented the methods by transferring only design commands in the agent-based environment. These methods give a good idea for Web-based design, but the limitations of environment and constraints reduce their applicability.

These methods help to realize Web-based modeling largely, but it is very hard to realize all the high-level modeling functions needed for complex product design. Indeed, this work is to develop a Web-based geometry engine alike. In fact, an effective model for transporting and displaying three dimensions product, is the key to realize cooperative deign. In order to overcome the problem that the product file in VRML format is always too big to be transferred and lack of high-level modeling functions, a powerful three dimensions solid model is needed to carry out product solid modeling effectively on the Web.

## 2 The Conceptual Solution

An effective CSCD system is to provide users with rapid model-displaying and convenient interactive functions. To archive this, two issues need to be addressed.

One is transporting dataflow between different clients. To answer user operations quickly, we need to decrease network data transferring during design. Moreover, a product model file is always too big to transport directly.



**Fig. 1.** In the RISM-based CSCD system structure, different users can operate same product cooperatively from different client applications in distributed environment

Another is the need of re-using the design result in manufacture processes. Therefore, the product model has to include not only geometry information but also modeling process information.

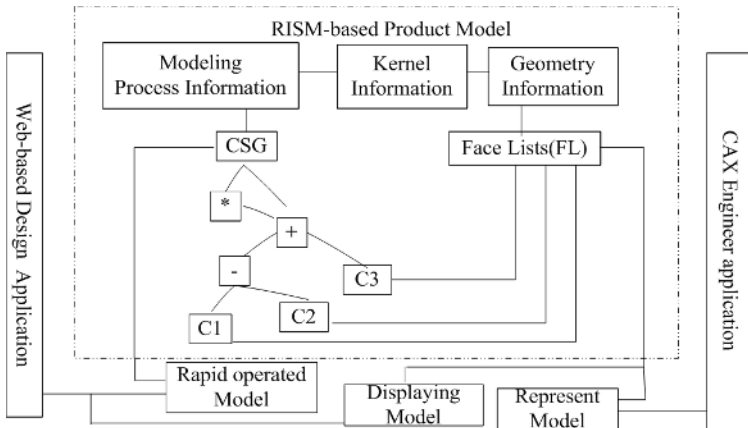
The goal of product design is to manufacture, so the design result needs a complete product representation. However, design process is a creative process and needs not all product information in detail. Therefore, the product model for design process is one kind of supporting “display model” in substance. This gives us an idea to solve data-transferring problem during cooperative design. If we divide the product geometry model into a “representation model” for manufacture and a “displaying model” for design process, the problem would change into how to build these two models and connect them. After dividing the two models, we can deal with design and manufacture processes separately. Then, we can take emphasis on the different requirements of design and manufacture. Based on this idea, the architecture of CSCD system is proposed as shown in Fig. 1.

Although the display model is mainly a simplified surface model for design use only, we can get a completed product model when all the operations back to server are carried out. By means of geometry engine such as ACIS, we could reconstruct or transform the model to CAM model.

### 3 RISM-Based Solid Modeling

#### 3.1 RISM Model

CSG is a traditional model representing CAD information completely and supports almost all the modeling operations. However, it is not enough for us to choose it as the main frame of the model for Web-based design. The CSG records the operations history and has similar tree-structure as VRML model. Therefore, a RISM (Rapid Induce Solid Model) is built by extending traditional CSG model. As shown in Fig. 2, RISM includes three parts of information.



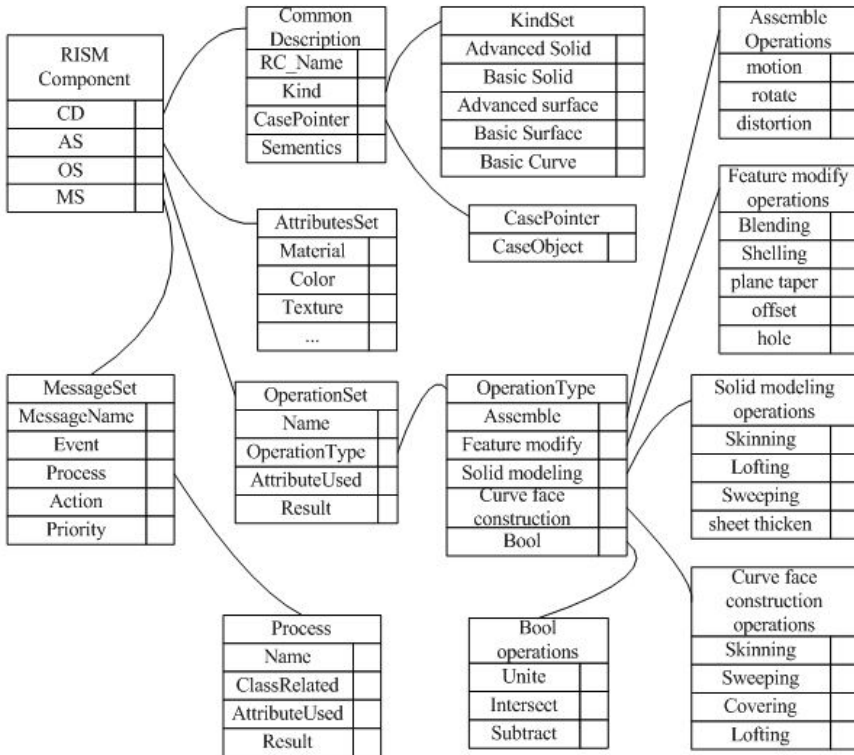
**Fig. 2.** In the architecture of RISM, C1, C2, C3 is Basic component of RISM. Ci could be basic curve, super curve-face, super solid and other kinds of RISM components.

Geometry information related to Face Lists (FL) contains the main information for the model displaying in Web design or generating true image. Modeling process information related to CSG records the product modeling process. It is essential to build a CSG-based operation history tree to answer rapidly. The kernel information saves common information such as kernel representation and other descriptions. Display model and representation model are connected by kernel information.

In RISM model, each FL of representation model relates to a component of CSG tree, that is, each FL has a certain relations with the component. Therefore, RISM divides product CAD model into two parts, displaying model for Web-based design, and representation model for engineering application. Moreover, displaying model and representation model could work individually and integrate into one body so as to build an effective solid model for Web-based applications.

### 3.2 Object-Based RISM Component

For the sake of computer based reasoning, we need a common structure to represent shapes. Based on RISM description, a formalized structure of RISM components is defined.



**Fig. 3.** In the description of RISM structure, CD is common information. AS is the attributes set which includes color, material, texture and so on. OS is the operations set in RISM Component. MS is the message set.

**Definition 1.** RC (RISM Component) is a four parameters group. It is expressed in the format like:

$$RC ::= \langle CD \rangle \langle AS \rangle \langle OS \rangle \langle MS \rangle \quad (1)$$

A complete description of RISM operations is shown in Fig. 3.

In fact, the product's operation history tree based on RISM, or the series of RISM operations, is the key factor for CSCD to model synchronously. Therefore, in most conditions, the displaying model in different clients could just transport some operation command rather than the whole product model on the networks.

RISM component contains more information than traditional CAD solid model. Using RISM component, a CSG tree could record the modeling process of a product or a part. If we call a case of RISM component as a RISM object, each product or part is treated as a RISM object, the attributes of products or parts can be represented by attributes of RISM object, products or parts modeling can be treated as operations of RISM object. Two patterns could be used to construct a completed product, one is a series of RISM operations in tree structure, and another is based on inherit network by ISA and Kind relations.

### 3.3 RISM-Based Design Process

When a message including objective RISM object and related operations is given, in theory, any two users can carry out cooperative design. By means of control mechanism of operations, a Cooperative Induce Manager (CIM) is built to manage displaying model in different clients. CIM manages cooperative work in two ways, one is to recognize user actions and to display results to the user, and another is to make related RISM objects of other users active by communicating with corresponding users. In order to explain how CIM works, several definitions are given as below.

**Definition 2.** Message Array is defined as message of cooperative design. One message array unit is called a MAU. A MAU is expressed in the format like:

$$MAU = \{RIO, RIM, User\} \quad (2)$$

In the expression, RIO represents a RISM object. In order to manage a product's inner RISM objects in tree structure, we generally attain the object close to the root when some operations are occurred so as to combine some operations. RIM expresses the RISM message transported, which includes the operations and some control parameters.

When RISM-based functional modules are built, design processes can be carried out. For all the design activities could be divided into a series of operations, one command is taken as an example and the processes are described as shown in Fig. 4.

Design commands from a designer are sent to the system. For the product organized by a CSG tree structure, the target RISM object would be a sub-tree of the whole product. And this object can be found from the CSG tree.

On the base of supported environment, which include the control strategy, geometry engineer and other functional process modules, the RISM object referred to common operations set is operated.

Then, two data sets will be sent back after the operations carry out, one is updated CSG sub-tree, and the other is the face lists of the RISM object. A LOD VRML file corresponding to the face lists is generated.

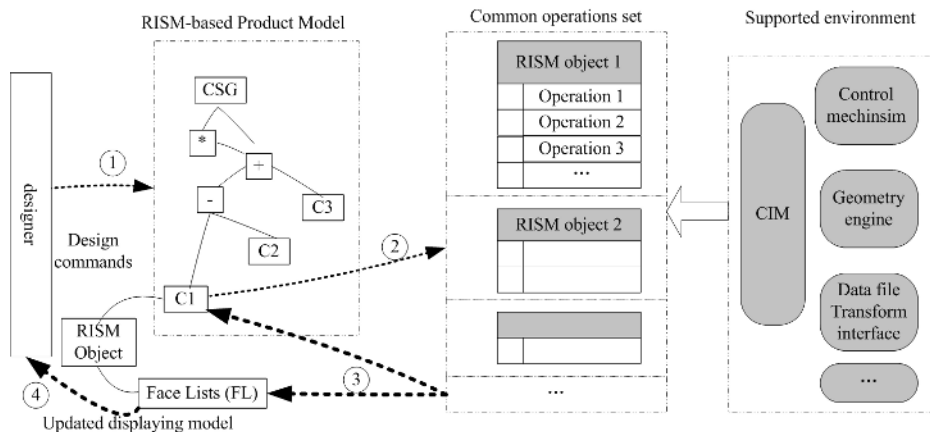


Fig. 4. Design process of RISM-based CSCW system

Finally, the VRML file is used to update the display model. The designer gets result back.

During the design process, CIM needs to manage conflicts when different users operate the geometry model. Some general rules are described as follows:

- Rule 1. Rule to eliminate a conflict MAU. When a CIM gets a conflict MAU, the check turn in MAU is RIO first, then RIM, User last.
- Rule 2. Rule to eliminate a conflict RIO. When a CIM gets a conflict RIO, it carries out the one RI object with high-rank in the CSG tree structure of product.
- Rule 3. Rule to eliminate a conflict RIM. When a CIM gets a conflict RIM, it carries out the one RI Message with a high priority.
- Rule 4. Rule to eliminate a conflict User. When a CIM gets a conflict User, it carries out the one with high-level user.

On the basis of common operations set and other functional module, the system can also find the modified parts of product and react to designers rapidly.

## 4 Development of a RISM-Based CSCD System

Based on the proposed RISM model, a prototype system is developed. We use Visual C++6.0 as a development tool and use ACIS 6.0 as geometry engine to develop server software. At the client side, we use JDK1.3 and JBuild7.0 to development corresponding Java Applets. The architecture of the prototype CSCD system is shown in Fig. 5.

- Design Interfaces get design commands from users and transform them into operation array back to server.
- Data Interactive Interface. The module exchanges data with design interface and generates display model for display interface. By means of LOD technology and other technologies, it enhances the client performance effectively.



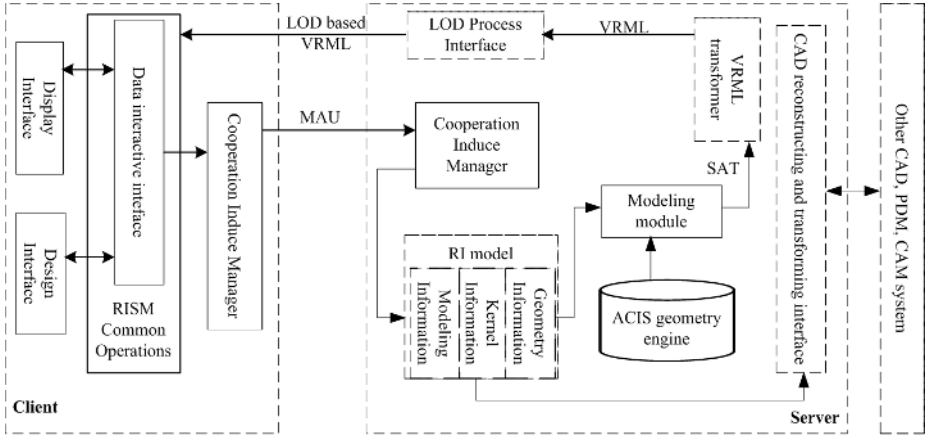


Fig. 5. RISM-based CSCW system structure is composed of eight function modules

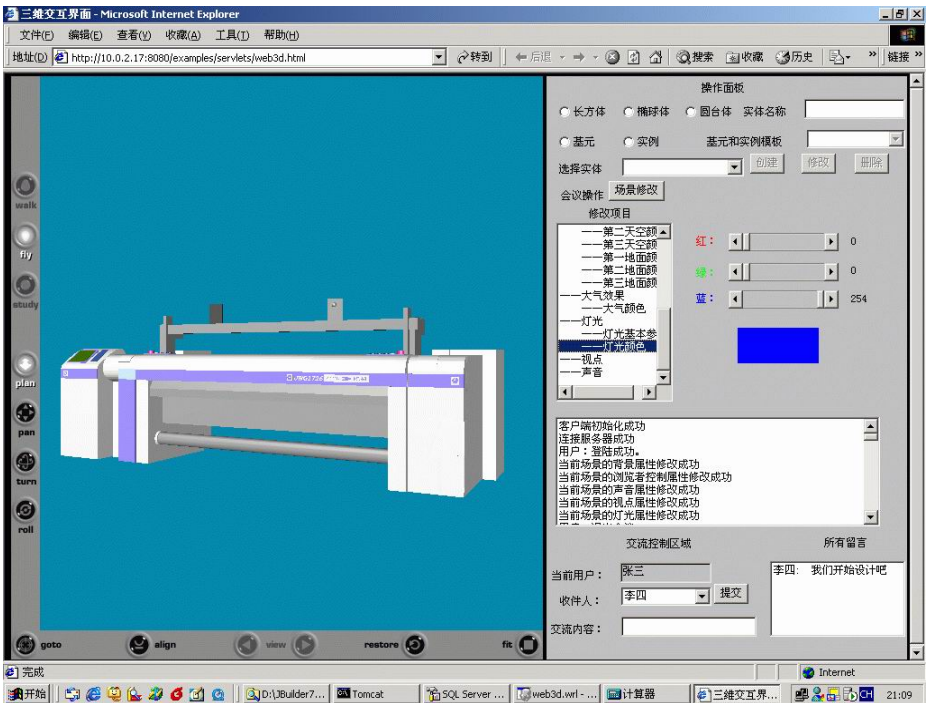


Fig. 6. Design view of CSCD client. Left is display area supporting by a VRML Plug-in. It provides users a virtual scene to design three dimension products interactively. Right up is operation area to answer user operation. A Java Applet realizes it, and it manages the cooperative meeting by CIM based on supporting bottom EAI interface. Right bottom is information area to provide user a communion place. It is realized as same as operation area.

- Cooperation Manager. The module builds and manages a cooperation meeting.
- RI Model. It is the kernel model of the system, and the solid model includes all the modeling information, kernel information and geometry information.
- Modeling Module. On the basic of ACIS geometry engine, it realizes modeling operation. And it transforms operations array from clients into SAT files.
- VRML Transformer. It transforms a product presented by B-rep model a SAT file into a VRML file.
- LOD Process Interface. The module processes VRML model to attain a LOD model so as to decrease data transporting in network.
- CAD Reconstructing and Transforming Interface. The module reconstructs result data or transforms RISM model into CAM model for manufacture.

Based on the ACIS geometry engine, RISM component is attained by encapsulating ACIS classes first. Then, we could call ACIS functions to model a certain product and get an object in an ACIS SAT file. At last, by means of triangle grid dividing and optimizing algorithm, a SAT file is transformed into a VRML file to display the model.

As shown in Fig. 6, the whole user interface is divided into three interactive areas.

The general process of interactive design can be described as follows. When the design operations of a user are got from a Web browser, it is explained as input events related to certain VRML nodes. Then EAI will construct an MAU data set and transfer it into supporting modeling module by Java Applet. According to the command back, the modeling system operates the product and transfers the modified part of product into an LOD VRML file. Then the Web browser updates when new VRML files are obtained.

## 5 Conclusion

A Rapid Induce Solid Model is proposed to decrease data-transporting so as to support Web-based interactive design. After building the basic RISM structure for representing shape and modeling operations, we extend traditional CSG model to build an operation history tree. Therefore, RISM component is provided to build CSG-based operation history tree. Then based on RISM common operations set and RISM message mechanism, a CSCD framework is built. In the proposed framework, the displaying model and representation model could work individually or together as an integrated system. At last, we implement a testing CSCD system. It shows that the RISM model provides a new and effective way to realize Web-based applications.

## Acknowledgement

This work was supported by the Fund of Scientific Research of The Open Key Laboratory on Railway Information Science and Technology of Railway Ministry and National 863 High Tech. Development Plan Fund of P.R. China (2003AA411310).

## References

1. Rischbeck, T., Watson, P.: A scalable, multi-user VRML server. Proceedings of Virtual Reality Annual International Symposium 2002, IEEE Computer Society, (2002) 199-206
2. Kiss, S.: Web based VRML modeling. Proceedings of Fifth International Conference on Information Visualization, (2001) 612-617
3. Hu, B., Tan, J.: A VRML-based product appearance customize method, Chinese Journal of Engineering Graphics, 2 (2002) 43-47
4. Hoppe, H.: View-dependent progressive meshes. Proceedings of Computer Graphics 1997 (SIGGRAPH'97), (1997) 189-198
5. Bidarra, R., van den Berg, E., Brosvoort, W.F.: Web-based collaborative feature modeling. Proceedings of sixth ACM symposium on solid modeling and applications, (2001) 319-332
6. Zhou, X., Li, J., He, F., Gao, S.: A Web-based synchronized collaborative solid modeling system. Chinese Computer Integrated Manufacturing Systems. 11 (2003) 960-965
7. Cai, H., He, Y., Wu, Y.: Construction of Extend Constructive Solid Geometry model for Web interactive design. Journal of Shanghai Jiaotong University (in Chinese), 12 (38) (2004) 2057-2062
8. Zhang, J., Zhang, S., Chen, C., Wang, B.: Distributed and synchronous collaborative design method on narrow bandwidth network. Journal of Shanghai Jiaotong University (in Chinese), 6(37) (2003) 882-886

# Author Index

- Abdulrahman, M.D. 156  
An, Yisheng 137
- Borges, Marcos R.S. 33, 45  
Brézillon, Patrick 45
- Cai, Hongming 448  
Cao, Hai 252  
Cao, Jian 270  
Chan, Mangtang 437  
Chen, David 230  
Chen, Deren 55  
Chen, Huowang 417  
Chen, Jihua 242  
Chen, Jiu Jun 95  
Chen, Yan 398  
Cheng, Zhong-Qing 427  
Chou, Shang-ching 299  
Cui, Lizhen 359
- Dai, Guozhong 11  
de Souza, Jano M. 117, 165  
Ding, Shuhui 175  
Dong, Jin-xiang 299  
Du, Xuan 398
- Fang, Min 368  
Fründ, Jürgen 289  
Fu, Xiangjun 199
- Gao, Ji 95  
Gao, Liping 105  
Gausemeier, Jürgen 289  
Ghenniwa, Hamada 127  
Gong, Bin 339  
Gong, Peng 252  
Guo, Chaozhen 147
- He, Yuanjun 448  
Heng, Xingchen 147  
Huang, Changqin 55  
Huang, Hong-Zhong 378  
Huang, Jiejun 86
- Ji, Peng 21  
Ji, Qingge 309  
Ji, Shuyan 211  
Jia, Fucang 319
- Li, Hua 319  
Li, Hui 252  
Li, Jiansheng 211  
Li, Renhou 137, 156  
Li, Shanping 199  
Li, Shaozi 417  
Li, Sikun 242  
Li, Xiu 279  
Liang, Lu 76  
Liao, Huaming 1  
Liao, Xiaoping 260  
Lin, Zongkai 252, 319  
Liu, Dazhi 175  
Liu, Hong 105  
Liu, Kecheng 437  
Liu, Ke-Jian 427  
Liu, Liyan 319  
Liu, Mei 175  
Liu, Shijun 339  
Liu, Shufen 67  
Liu, Weiwei 221  
Liu, Wenhua 279  
Liu, Xiyu 105  
Luo, Junzhou 21
- Ma, Huadong 349  
Ma, Hui 76  
Mao, Qirong 406  
Matysczok, Carsten 289  
Medeiros, Sergio P. 165  
Meng, Xiangxu 339  
Miao, Jian 260
- Palma, Sérgio 117  
Pan, Heping 86  
Pan, Yan 76  
Pan, Zhigeng 309  
Peng, Chenglian 398  
Pinho, Daniel 117

- Pino, Jose Alberto 45  
 Pomerol, J.-Ch. 45  
  
 Radkowski, Rafael 289  
 Raikundalia, Gitesh K. 328  
  
 Santoro, Flávia Maria 33  
 Santos, Neide 33  
 Schneider, Daniel S. 165  
 Shen, Weiming 127  
 Sheng, Song En 95  
 Shi, Dongping 252  
 Su, Daizhong 211  
 Sun, Linfu 187  
 Sun, Zhaoyang 175  
  
 Tang, Min 299  
 Tang, Na 76  
 Tang, Yong 76  
 Tian, Feng 156  
  
 Vivacqua, Adriana 117  
  
 Wan, Youchuan 86  
 Wang, Haiyang 359  
 Wang, Hui 11  
 Wang, Jinfeng 406  
 Wang, Shaorong 319  
 Wang, Xianqing 55  
 Wang, Xiaozhi 21  
 Wang, Ying Daisy 127  
 Wu, Jia 147  
 Wu, Minghui 368  
 Wu, Xiao-Qiang 388  
 Wu, Yong 448  
  
 Xexéo, Geraldo B. 165  
 Xiang, Hui 339  
 Xiong, Zhihui 242  
 Xu, Bing 309  
 Xu, Zhiwei 1  
  
 Yan, Jun 328  
 Yang, Hongwei 309  
 Yang, Ning 1  
 Yang, Yun 328  
 Yao, Zhilin 67  
 Ye, Feng 270  
 Ye, Lu 309  
 Ying, Jing 368  
 Ying, Zheng-ming 299  
 Yu, Zhen-Wei 427  
  
 Zeng, Qinghuai 55  
 Zhan, Yongzhao 406  
 Zhang, He 279  
 Zhang, Heming 230  
 Zhang, Jincheng 156  
 Zhang, Maojun 242  
 Zhang, Xinfang 260  
 Zhang, Xinjia 67  
 Zhong, Peisi 175  
 Zhou, Changle 417  
 Zhou, Gengui 270  
 Zhou, Laishui 221  
 Zhou, Xuegong 398  
 Zhu, Ye 21  
 Zhu, Zhiting 55  
 Zhuang, Haijun 221  
 Zu, Xu 378