LNAI 3891

Jaime S. Sichman
Luis Antunes (Eds.)

Multi-Agent-Based Simulation VI

International Workshop, MABS 2005
Utrecht, The Netherlands, July 2005
Revised and Invited Papers

Springer

Lecture Notes in Artificial Intelligence 3891

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Jaime S. Sichman   Luis Antunes (Eds.)

Multi-Agent-Based Simulation VI

International Workshop, MABS 2005
Utrecht, The Netherlands, July 25, 2005
Revised and Invited Papers

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Jaime S. Sichman
Universidade de São Paulo
Escola Politécnica
Depto. Eng. Computação e Sistemas Digitais (PCS)
Laboratório de Técnicas Inteligentes (LTI)
Av. Prof. Luciano Gualberto, 158, trav. 3, 05508-970 São Paulo, Brazil
E-mail: jaime.sichman@poli.usp.br

Luis Antunes
Universidade de Lisboa
Faculdade de Ciências
Grupo de Estudos em Simulação Social (GUESS)
Campo Grande, 1749-016 Lisboa, Portugal
E-mail: xarax@di.fc.ul.pt

Library of Congress Control Number: 2006923044

CR Subject Classification (1998): I.2.11, I.2, I.6, C.2.4, J.4, H.4

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN	0302-9743
ISBN-10	3-540-33380-0 Springer Berlin Heidelberg New York
ISBN-13	978-3-540-33380-7 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11734680 06/3142 5 4 3 2 1 0

Preface

This volume groups together the papers accepted for the 6th International Workshop on Multi-Agent-Based Simulation (MABS 2005), co-located with the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), which occurred in Utrecht, The Netherlands, on July 25, 2005.

MABS 2005 is the sixth workshop of a series that began at ICMAS 1998 (Paris, France), and continued successively with ICMAS 2000 (Boston, USA), AAMAS 2002 (Bologna, Italia), AAMAS 2003 (Melbourne, Australia) and AAMAS 2004 (New York, USA). The revised version of the papers of these workshops appeared in Springer's *Lecture Notes in Artificial Intelligence,* in volumes 1534, 1979, 2581, 2927 and 3415. All information about the MABS Workshop Series can be found at `http://www.pcs.usp.br/~mabs`.

After some hesitations about the numbering of the volumes, we decided to set the pace right between the workshop edition and the volume name. So, this volume is called *Multi-Agent-Based SimulationVI*, and subsequent editions of the book series will correspond to the ordinal number of the workshop.

The scientific focus of MABS lies in the confluence of social sciences and multi-agent systems, with a strong applicational/empirical vein, and its emphasis is on (i) exploratory agent-based simulation as a principled way of undertaking scientific research in the social sciences and (ii) using social theories as an inspiration to new frameworks and developments in multi-agent systems.

As the area of agent-based simulation is quickly auto-organizing, MABS has proposed itself as a forum for social scientists, agent researchers and developers and simulation researchers to assess the current state of the art in the modeling and simulation of social systems and multi-agent systems, identify where existing approaches can be successfully applied, learn about new approaches and explore future research challenges.

MABS 2005 attracted a total of 28 submissions from 14 different countries (Belgium, Brazil, Canada, France, Germany, Hungary, India, Japan, Portugal, Spain, The Netherlands, Tunisia, UK, USA). Every paper was reviewed by three anonymous referees, and in the end 12 papers were accepted for presentation in the workshop. Every paper was later reviewed again by a Program Committee member for this volume.

We are very grateful to every author who submitted a paper, as well as to all the members of the Program Committee and the additional reviewers for their hard work. The high quality of the papers included in this volume would not be possible without their participation and diligence. We would also like to thank Scott Moss and Emma Norling, who gave a very wholehearted invited talk.

Thanks are also due to Rino Falcone (AAMAS 2005 Workshop Chair), to Sarit Kraus and Munindar Singh (AAMAS 2005 General Chairs), to Sven Koenig and Michael Wooldridge (AAMAS 2005 Program Chairs) and to Frank and

Virginia Dignum (AAMAS 2005 Organization Chairs). Finally, we would like to thank Springer staff, especially Alfred Hofmann and Christine Günther, for their support of MABS and their help in the making of this book.

Social simulation is a terrifically exciting scientific endeavor, and MABS contributes to the significant role of bridging this multidisciplinary area with computer science and especially multi-agent systems.

January 2006

Jaime Simão Sichman
Luis Antunes

Organization

Organizing Committee

Luis Antunes (Universidade de Lisboa, Portugal)
Jaime Simão Sichman (Universidade de São Paulo, Brazil)

Program Committee

Frédéric Amblard (Université Toulouse 1, France)
Luis Antunes (Universidade de Lisboa, Portugal)
François Bousquet (CIRAD/IRRI, Thailand)
José Castro Caldas (ISCTE, Portugal)
Cristiano Castelfranchi (ISTC/CNR, Italia)
Helder Coelho (Universidade de Lisboa, Portugal)
Rosaria Conte (ISTC/CNR, Italia)
Noshir Contractor (University of Illinois at Urbana-Champaign, USA)
Luis Correia (Universidade de Lisboa, Portugal)
Paul Davidsson (Blekinge Institute of Technology, Sweden)
Jim Doran (University of Essex, UK)
Tom Dwyer (UNICAMP, Brazil)
Alexis Drogoul (IRD, France)
Nigel Gilbert (University of Surrey, UK)
Wander Jager (University of Groningen, The Netherlands)
Marco Jansen (Arizona State University, USA)
Jorge Louçã (ISCTE, Portugal)
Juan Pavón Mestras (Universidad Complutense Madrid, Spain)
Scott Moss (Manchester Metropolitan University, UK)
Jean-Pierre Müller (CIRAD, France)
Emma Norling (Manchester Metropolitan University, UK)
Mario Paolucci (ISTC/CNR, Italy)
Juliette Rouchier (Greqam/CNRS, France)
Keith Sawyer (Washington University in St. Louis, USA)
Jaime Simão Sichman (Universidade de São Paulo, Brazil)
Carles Sierra (IIIA, Spain)
Ron Sun (Rensselaer Polytechnic Institute, USA)
Keiki Takadama (Tokyo Institute of Technology, Japan)
Oswaldo Teran (Universidad de Los Andes, Venezuela)
Jan Treur (Vrije Universiteit Amsterdam, The Netherlands)
Klaus Troitzsch (University of Koblenz, Germany)
Afzal Upal (University of Toledo, Ohio, USA)
Harko Verhagen (Stockholm University, Sweden)
Gérard Weisbuch (Ecole Normale Supérieure, France)

Additional Reviewers

Diana F. Adamatti (Brazil)
Luciano Reis Coutinho (Brazil)
Andrea Giovannucci (Spain)
Mark Hoogendoorn (The Netherlands)
Francesca Marzo (Italy)
Luis Moniz (Portugal)
Alexei Sharpanskykh (The Netherlands)
Yutaka Leon Suematsu (Japan)
Paulo Urbano (Portugal)

Table of Contents

Environments

Multi-Agent-Based Simulation: Why Bother?

Scott Moss and Emma Norling

Centre for Policy Modelling, Manchester Metropolitan University
{scott, emma}@cfpm.org

Abstract. This year's MABS workshop was the sixth in a series which is intended to look at "using multi-agent models and technology in social simulation," according to the the workshop series homepage [1]. We feel that this is an appropriate time to ask the participants and the wider community what it is that they hope to gain from this application of the technology, and more importantly, are the tools and techniques being used appropriate for achieving these aims? We are concerned that in many cases they are *not*, and consequently, false or misleading conclusions are being drawn from simulation results. In this paper, we focus on one particular example of this failing: the consequences of the inappropriate use of numbers. The translation of qualitative data into quantitative measures may enable the application of precise analysis, but unless the translation is done with extreme care, the analysis may simply be more precisely *wrong*. We conclude that as a community we need to pay careful attention to the tools and techniques that we are using, particularly when borrowing from other disciplines, to make sure that we avoid similar pitfalls in the future.

1 Introduction

The Multi-Agent-Based Simulation workshop series began in 1998, with the stated aim of developing "stronger links between those working in the social sciences, for whom agent-based simulation has the potential to be a valuable research tool, and those involved with multi-agent simulation, for whom the social sciences can provide useful theories and exemplars" [2]. Over the years, the community has developed and expanded, with participants from all corners of the globe, and applications in a wide range of areas. How strong are the links though, between those working in social sciences and those working in multi-agent simulation? Recent workshops have been dominated by researchers with backgrounds in agent-based simulation, with little input from the social scientists. Is that simply because of the workshop's association with the AAMAS conference series (which contains little else of interest to social scientists), or is it a symptom of a more serious problem: that the work of those in the MABS community is seen as largely irrelevant to those in the social science community? We do not believe that we have quite reached this critical situation, but fear that we have identified some disturbing trends that are leading us as a community down this path.

Rather than pointing directly to examples of these trends in MABS research, we illustrate our point here with examples from a related field, management

J.S. Sichman and L. Antunes (Eds.): MABS 2005, LNAI 3891, pp. 1–13, 2006.

research, that is already regarded with scepticism by many management practitioners. We believe that there are strong parallels between the examples from this field and the research presented in MABS. Not only are there commonalities in the topics studied (for example, self-managing groups [3, 4] and innovation [5, 6]), but similar techniques are used to interpret and analyse data. The claim has been made that management education is irrelevant – or even downright harmful – by Pfeffer and Fong [7] and the late Sumantra Ghoshal [8], both writing in the Academy of Management's *Learning & Education* as well as, in the practitioner press, Bennis and O'Toole [9] in the *Harvard Business Review.* While the tag line for these articles claims a crisis in business education, they have in common a critique of academic research and scholarship in the area of management. In this paper, we aim to show firstly that practicing managers are right to doubt the relevance of leading management research as represented by articles published in recent issues of the most influential journals in the field.

The critique of current management research follows from the following dictum: formalism confers precision but neither descriptive nor predictive accuracy. The desire for precision amongst management researchers is evidenced by the practice of summarising notions and concepts from secondary evidence as propositions or hypotheses. Frequently, these propositions or hypotheses are then tested by some statistical means or restated as algebraic formulations. There is a question as to whether particular formal or pseudo-formal formulations produce any descriptively or predictively accurate statements and also whether statistical tests devised for these statements are able to distinguish what is accurate from what is simply precise.

In addition, precision itself has a cost even where precise formulations are also accurate. The cost is in terms of expressiveness. An algebraic expression or a regression model denote only what can be expressed precisely by numbers. Formal logics can use mnemonic terms in the drawing of inferences from axioms and rules of inference. The mnemonic terms are more expressive than number and, in general, are more expressive the more closely they resemble natural language. Natural language, of course, is rich in connotation but much less precise than formal logics.

As a community, we should be asking ourselves *"What is it that we hope to achieve with our simulations, and are the tools and techniques that we are using appropriate for achieving these aims?"* The second part of this question cannot be answered without answering the first, and so in the next section of this paper we consider why we do simulation. We also need to understand the intrinsic characteristics of our simulations (Sect. 3), as well as the limitations and bounds of the tools and techniques that we choose to use. In this paper, we do not intend to analyse all the tools and techniques that are used in our field. Rather, we show how one technique – the translation of qualitative data into quantitative data – is often mis-used (Sect. 4). We draw our examples from management research rather than the MABS community, but the techniques used in the cited examples are also used in social simulation, and the same pitfalls must be avoided. We present a discussion of how we believe agent-based simulation can contribute to these particular types of problems (Sect. 5), and in summation urge all researchers to reconsider their tools and techniques, particularly when 'borrowing' from other disciplines, to ensure that they are being applied in an appropriate way. If not, we run the risk of presenting irrelevant, if not invalid, conclusions and having little impact with our research.

2 Why Do We Do Simulation?

Good science enables us to understand what we observe. Different sciences have different criteria of what it means to do this. In physics, the depth of understanding is judged on the prediction of specific events and phenomena or distributions of numerical measures. Evolutionary biologists do not predict the emergence of previously described species but they do provide an explanation of speciation that was developed by Darwin to cohere with the fossil record and has subsequently cohered with statistical and molecular genetics. We assume that researchers in our field also wish to contribute to the advancement of science; the question then becomes, how do we see our simulations to be doing this?

For social simulation, the main area of interest in this workshop series, the processes of interest are social processes: the way that people interact, and the effect that this has upon society (or a sub-group of society) as a whole. The predictive powers of our simulations depend upon the explanatory powers of earlier simulations: until we have satisfactorily produced explanations of behaviour, we can have little confidence in predicting future behaviour. For this reason, many of the simulations that we develop are intended to test theories, implementing the theories to see if the expected results are produced. In this year's workshop, an example of this use of simulation that of of Antunes et al. [10] who are seeking to understand tax compliance. Other simulations are intended to be used predictively, particularly for evaluating how proposed new policies or systems will affect a system's behaviour. Melo et al's police patrol route simulation (also in this year's workshop) gives an example of this [11].

Another type of simulation that is discussed in this workshop series is that which draws upon social science for inspiration in the algorithms used for distributed systems. These simulations are typically used to evaluate the algorithms, comparing them against alternatives. Examples of this type of simulation that appeared in this year's workshop are those of Rodrigues and Luck [12] and Sultanik et al. [13]. This paper is more relevant to those who are interested in social simulation than this 'socially-inspired' simulation, because we are particularly focusing on how human knowledge is represented. However the general lesson – that researchers should take care in the application of tools and techniques, and understand their limitations – applies equally here.

3 The Nature of Social Simulation

In social simulation, much of the data that we are capturing is human knowledge: how people make decisions, relate to others, etc. In some cases, this knowledge is naturally quantitative (for example M has 5 children) but more often than not, it is qualitative (for example, X has a business relationship with Y, or A is more thrifty than B). The problem that we face is that more often than not, the formalisms (that is, programming languages) that we use more naturally represent quantitative data than qualitative data, and so we have a tendency to transform from latter to the former. While there is nothing inherently wrong with translating qualitative data into quantitative, this process can introduce serious errors into the data and its analysis, as will be seen in the examples in Sect. 4.

By using declarative languages (or languages with declarative features) in our simulations, we can avoid the need to translate qualitative data into numerical data. However this alone will not free us from troubles with numbers: almost certainly there will be some numerical – probably statistical – analysis of the simulation results, and it is equally important then to understand the assumptions underlying and limitations of the techniques being used. We are not the first to have made this point – our colleague Bruce Edmonds is one who has repeated argued for care in numerical representation and analysis [14, 15]. We stress again that although our examples focuses on one particular type of problem, this problem is intended to be representative, rather than comprehensive. There are other examples of assumptions or limitations that are ignored, and these too lead to questionable inferences and conclusions.

4 A Case in Point: Management Research

As previously mentioned, we take our examples from management research rather than pointing directly at examples in the MABS literature. As academics in a business school, our research enters into both social science and management research, using agent-based-simulation as a tool. Here we identify some of the weaknesses in management research; in the following section we consider the implications of this for our research community.

It should be noted that although strong arguments against theoretical management research have been presented, as discussed in Sect. 1, the management research community as a whole do not acknowledge this gap between theory and practice. The establishment view on the principled relationship between theory and evidence is taken from the mission statements of leading, elite, world class journals of management research: *The Academy of Management Review*, *The Academy of Management Journal*, *The Strategic Management Journal* and *Administrative Science Quarterly*. These journals represent four of the top five journals[1] in the field of management ranked by their impact index in the ISI Journal Citation Reports. The journal *Management Science* is also considered to give us the top five journals in the field ranked by number of citations to their articles and because it is far and away the most frequently cited journal.

[1] *MIS Quarterly* was not included because it is more specialised than the others and therefore does not necessarily reflect the most general conventions in management research.

Table 1. Mission statements of five top management research journals

Academy of Management Review	[E]ach manuscript published in AMR must advance theory or the theory development process in the area of management and organizations. [16]
Academy of Management Journal	The mission of the Academy of Management Journal is to publish empirical research that tests, extends, or builds management theory and contributes to management practice. [17]
Strategic Management Journal	It is devoted to the improvement and further development of the theory and practice of strategic management and it is designed to appeal to both practising managers and academics.[18]
Administrative Science Quarterly	The ASQ logo reads, "Dedicated to advancing the understanding of administration through empirical investigation and theoretical analysis." ... Theory is how we move to further research and improved practice. If manuscripts contain no theory, their value is suspect. Ungrounded theory, however, is no more helpful than are atheoretical data. [19]
Management Science	Management Science seeks to publish articles that identify, extend, or unify scientific knowledge pertaining to management. [T]he unifying thread ... is a fundamental focus on improving our understanding of the practice of management. Within this scope, theoretical, empirical, prescriptive and descriptive contributions are welcome. [20]

While one of these mission statements – that of the *Academy of Management Review* – does not mention empirical work, even in that case arguments are based on secondary evidence. In general, it seems clear that there is an establishment view that in practice evidence is an essential aspect of good management research. This view is very different from (say) that of the economics establishment where the most highly prized articles are frequently those that build some new theoretical insight on top of some previous theoretical basis that itself could hardly be more divorced from any reality we have ever observed. In other words, management research ideal is closer to the best of the natural sciences in its regard for evidence that is the norm in the social sciences. Of course, this leaves open the question of how the ideal translates into practice – a question to which we now turn.

From the journals mentioned in Tab. 1, we present a detailed analysis of two articles from issues in early 2005. These two articles have been selected not because they are in any way special, merely that they are two typical articles from the leading journals at this particular point in time. Analysis of other articles from the same pool reveals many similar features; there is simply not sufficient space to report this analysis here.

4.1 A First Example

The standard procedure in management research is to present a discursive verbal account of published literature and observational (as distinct from statistical) evidence in order to formulate propositions or hypotheses. At the extreme theoretical end of the spectrum, we consider the general articles in the most recent (at the time of writing) issue of *Academy of Management Review* (AMR). Of the three general articles in this issue (the remainder being devoted to a special topic forum, *Do Governments Matter?*), all followed a virtually standard template. They review prior theoretical and empirical literature. They devise on the basis of their literature reviews a set of propositions which then form the basis of a set of inferences for organisational management.

Taking the second of these three articles for detailed analysis, Hackman and Wageman look at the type and timing of coaching work teams during the course of a task or project with a well-defined duration [21]. The literature surveyed to formulate four propositions was drawn from psychology, organisation behaviour, social psychology (small group theory) and practitioner literature on team management. It is instructive here to follow selected links in the literature chain on which Hackman and Wageman relied. One supporting element in their argument is earlier work published by Wageman [22].

In that article, Wageman constructs a measure of the level of self-management of a work team by taking the arithmetic mean of the three indices for collective responsibility, managing and monitoring the team's own performance. These three individual measures are themselves constructed as follows:

> Three theory-specialized components of self-management (collective responsibility for work outcomes, monitoring own performance, and managing own performance) were coded separately (as high, medium, or low) and then averaged to form an overall measure of a team self-management (Cronbach's alpha=0.94) [22, p. 566].

These measures are then used to compare the effects of 'negative leader coaching' and 'positive leader coaching', producing graphs that appear to argue convincingly for the latter [22, p. 571].

Cronbach's alpha is mathematically equivalent to taking every possible split of the data into two halves and calculating the correlation coefficient between each of these split halves and then the average of those correlation coefficients [23]. It is a non-parametric test so there are no further tests of the goodness of fit of Cronbach's alpha. Nonetheless, a value of 0.94 is unusually high – the SPSS manual states that in the social sciences values above 0.8 are considered reliable.[2] The orthogonality of the measures on the two axes is deduced from a principal components analysis.

Both Cronbach's alpha and principal components analysis depend on manipulations of correlation matrices. The correlation coefficients are obtained by taking differences of observations from means, multiplying such differences and dividing through. The measures for which the correlation coefficients were obtained in this case are themselves arithmetic means where the observations being averaged are of three different kinds. The observations take values low, moderate

[2] We will not dwell on the Cronbach alpha values of 0.77 for "positive coaching" and 0.56 for "negative coaching" [22, p. 568].

and high which are given arbitrary values of 1, 2 and 3, respectively. Low, moderate and high are naturally a partial ordering. High is greater than moderate which in turn is greater than low.

Wageman [22] calculated the standard inter-rater reliability (IRR) to determine if the different interviewers of managers and work team members were consistent in their assignments to these levels. That they were (the IRR ranged from 0.92 to 0.95), indicates that the partial orderings were sensible and consistent. That does not make it reasonable to assert that one high (=3) is the same as one low (=1) plus one moderate (=2) or that the ratio of high to moderate is 1.5 while the ratio of moderate to low is 2. Since the measures themselves imply the validity of such operations, the whole statistical analysis stands or falls with the cardinality of low, moderate and high. Moreover, the correlation matrix is invariant with respect to linear, but only linear, transformations. Consequently, the orthogonality of the measures and the appropriate components of those measures cannot be assumed unchanged if, for example, the values of low, moderate and high are all raised to some power. If the exponent is less than one, the correlation coefficients will be raised. If the exponent is greater than one, the coefficients will be lowered. The relative correlation coefficients will change so that the measures of orthogonality and consistency will change. Suppose, for example, that just as moderate is twice the value of low, high is twice the value of moderate. All correlation coefficients will be reduced, the value of Cronbach's alpha will be lowered (giving less credence to the appropriateness of combining the particular set of components of the self-management and the leader coaching indices) and might well (but need not) change the indications of orthogonality from the principal components analysis.

The more general issue raised by this problem is the link between essentially qualitative data and formal analytical techniques. It is possible that statistical techniques are not the natural class of formal analytical techniques for at least some issues in management research – or indeed social simulation. The formalisation of qualitative data as statistics might be sufficiently unnatural that its relevance is hard for practicing managers to see.

In order to assess the generality of the issue of qualitative data of the sort which appears to be a mainstay of management research and the sort of statistical analysis that is sometimes applied to it, we look at another paper in Hackman and Wageman's citations. Cohen, Ledford and Spreitzer [24] use statistical techniques with the same reliance on correlation coefficients as do Hackman and Wageman. The main difference is that Cohen et al. use factor analysis to provide the structure for structural equations modelling while Hackman and Wageman use principal components analysis to reduce the number of factors in their analysis. Both are standard statistical techniques used in standard ways. *And both papers rely on the cardinality of the Likert scales up to a linear transformation.*

To relate the issue here to the general question of the relevance of management research to management practice, we must ask whether it is reasonable for practicing managers to question the relevance of research that depends on whether the difference between moderate and high is unambiguously equal to the difference between moderate and low when what is being measured is "level of self-management" or "positive team coaching" or, from Cohen et al., "trust", "commitment", "self-criticism" and similar phenomena. None of these phenomena have any natural numerical measure. In their choice of statistical techniques, these contributors to the literature on self-managing teams have gone beyond what is formally consistent with the partial orderings implied by their data. Without some further justification for the use of the techniques these authors have chosen, their statistical results have no clear formal basis. That is, there is no formal scientific reason to accept these results as valid and the authors do not offer any evidence or even argument for their plausibility. In the circumstances, it would seem hard to argue with a practicing manager who was unwilling to rely on these results for structuring an organisation or, in particular, choosing between self-managed and hierarchically managed work teams.

4.2 A Second Example

Matters are not essentially different when we turn to the less theoretical journals such as the *Academy of Management Journal* (AMJ). In the first article in the April 2005 issue, Milton and Westphal are concerned with the effect on group performance of having individuals share one another's views of their personal qualities [25]. Drawing on both social psychology and prior research on work groups, they offer propositions on pairs of individuals within the groups and also about the roles of individuals within the group as a whole. Once again, Likert scales are used. In this article, the nine-point scales are applied to attributes such as "likeable in general," "competent in most things," "goes with the flow," "is understanding of others," "is warm to others." Identity confirmation was measured on 16 such attributes "by computing the absolute value of the difference between the dyad partners rating of the focal member on a dimension and the focal individual's self rating on the same dimension" [25, p. 196]. Suppose that two individuals both rate one of them at '6' on the nine-point Likert scale for "likeable in general". Suppose that a third individual rates one of the first two (the focal individual) at '7'. We suppose yet further that each individual has a partial ordering on likeability over all other individuals in the work group. The '7' of one of these individuals and the six of the other non-focal individual could, in principle, amount to the same ranking of the focal individual amongst all individuals in the group. In that case, in what way is the '6' of one different from the '7' of the other? Without some clear indication of what that means, there is no sense to be made of the claim that the non-focal '6' is closer than the non-focal '7' to the focal '6'.

The authors buttressed the claim to the validity of their procedure by appealing to a set of constraints on difference scores specified by Edwards [26, 27]. This involved regressing "cooperation" on the difference scores for each of the 16 attributes. Cooperation was defined as "interactive and relational behavior that occurs between members of a work group and that is directed at task achievement in the group." The measure of cooperation was the usual mean of Likert scale values for three behavioral characteristics. A seven-point Likert scale was used for the cooperation measure obtained by getting every participant in a work group to rate every other person on how much they relied on them, how much they discussed problems and concerns with them and how much conflict they experienced with them. The regression equation was

$$Z = b_0 + b_1 X_i + b_2 Y_i + b_3 W_i + b_4 W_i X_i + b_5 W_i Y_i + e \quad (1)$$

For individuals A and B, Z is "cooperation", X_i is the Likert scale value difference between A's self assessment and B's assessment of A's attribute i. Y_i is the difference between B's self assessment and A's assessment of B's attribute i. This equation is equivalent to regressing Z on the absolute differences of the assessments of each individual's attribute i if W_i is a dummy variable taking the value 1 when $X < Y$ and 0 when $X \geq Y$ and, in addition, $b_1 = b_2$, $b_4 = b_5$, $b_3 = 0$ and $b_4 = 2b_1$ [27, p. 357]. Since these are regression coefficients, Edwards argues that an attribute should only be used if the calculated coefficients are not statistically significantly different from the values indicated by the above constraints. However, as the calculation of regression coefficients turns on correlation coefficients, we again have the unstated requirement that Likert scales be linear.

4.3 The Use of Likert Scales

Performing regression and cluster analyses on Likert scale data is a standard practice in management research and is widespread even in the best journals. Our claim is not that it is necessarily wrong but that there is a universal (and universally unstated) assumption that any linear Likert scale is an appropriate representation of the set of qualitative judgements available to individuals. If that assumption is not satisfied then statistical analyses using Likert scale data might or might not be misleading.

If management research is to follow the most successful sciences in theory constrained by evidence, then some such constraint is needed in determining how best to formalise the qualitative data currently formalised by Likert scales. The first question we ask is whether a Likert scale or something similar to a Likert scale is an appropriate formalisation of the qualitative information obtained by survey, interview, focus groups or any other means.

The Likert scale entails a comparison of the qualitative attributes of objects, ideas, situations, or other tangible and intangible phenomena. Consider any of these phenomena as conceptual objects. It is possible to ask informants whether they would prefer a particular instance of an object or some other instance. If the several instances of the object cannot be had simultaneously – for example, a situation – then one can ask informants whether they would prefer one or another length of sequences of such objects.

Now consider the following thought experiment:

We have a set of objects (which need not be tangible) to which we can assign attributes such as "very important", "important", "not very important", "rather unimportant" or "unimportant". These categories can be given numerical values from 1 (for "unimportant") to 5 (for "very important"). In our thought experiment, we ask *How many objects with the attribute "important" would be said by a specified individual to be collectively as important as some number of objects with the attribute "very important?"* In principle we could ask this or some equivalent question to determine empirically how much more valuable is the attribute "very important" than is the attribute "important". If a Likert scale was an appropriate measure then $5x = 4y$ where x is the number of "very important" objects and y is the number of "important" objects. The virtue of this procedure is that it can be tested by the same means that management and other social researchers use to get the qualitative evidence that they assume can be processed by means of some instance of the general linear model. In the present case, $y = 1.2x$: the number of objects of type y must be 1.2 times the number of objects of type x for the individual to value each collection equally. Similarly, the Likert scale requires that the individual be indifferent between two "unimportant" objects and one "not very important" object or between three "not very important" and two "important" objects, and so on. This will yield a consistent linear scale in the interval from 1 to 5 inclusive.

If we linearly transform this scale, then as we have seen there will be no effect on the results from using any instance of the general linear model. But it will change the equivalence relations among the numbers of objects of each attribute. For example, if we transform the scale so that $x' = 2 + 3x$ then five "unimportant" objects will be of the same value to the individual as one "rather unimportant" object and 17 "unimportant" objects will be of the same value to the individual as one "very important" object. There will be no difference in the correlation coefficients or, therefore, the values of the Cronbach alpha or the results of principal components or factor analyses. There will, however, be differences in regression coefficients and the goodness-of-fit tests. Consequently, even a linear transform of the the Likert scale is likely to have an impact on the analysis of the resulting data.

Other – that is, non-linear – transforms are also plausible a priori. Suppose that, for some individuals, two "unimportant" objects are of the same value as one "rather unimportant" object, two "rather unimportant" objects have the same value as one "not very important" object, and so on through the whole scale. That is, for each category, an object has the same value as two objects characterised by the category below. Then the transformed value of any Likert scale point is

$$v_i = be_i \tag{2}$$

where e_i is is the index of the point on the Likert scale, b is the number of objects of one category (e.g., "important") that are collectively of equal worth to the individual as one item of the next category up (e.g., "very important") and v_i is the transformed value. This is a very flexible transform since b can take any value greater than 1 and it is not necessary to use every integer in sequence. Indeed, the e_i values need not be integers. Thus, if b is close to 1, then differences between the contiguous categories on the Likert-type scale can allow for gradations as small or as large as is wanted and, moreover, the difference between (say) "very important" and "important" can be larger or smaller than the differences between any other two contiguous categories. The valuations remain as consistent across categories as in the linear case but, as the transformed scales are not linear, the general linear model is inapplicable to the data.

The difficulty is compounded if different individuals' concerns, interests and preferences are best represented by different values of b and e_i. Moreover, we have only weak and casual evidence that the transform given by (2) is itself an appropriate representation of how individuals value the tangible, the intangible and the situational. Another possibility is that individuals rank alternatives lexicographically as in Cohen [28]. That is, they would choose an object with the most attributes at the highest ranking. In the example developed here, if an object had three attributes labeled "very important" and no other object had as many, then the first object would be chosen over all of the others even if they had many "important" attributes.

Of course, Likert scales are not usually used explicitly as a representation of a choice algorithm. The thought experiment devised here does represent the scales as a choice algorithm in order to devise a procedure for assessing the nature of the scale – in particular whether the linearity assumption is supported by independent evidence. Nonetheless, scholars who use Likert scales are implicitly assuming that they indicate the reasons why decision-makers choose as they do and, so, even in those cases the Likert scales are not divorced from choice criteria.

5 The Implications for Multi-Agent-Based Simulation

What then are the implications of these analyses for our field of research? Most obviously, care must be taken when translating what is inherently qualitative data into quantitative data. There is no doubt that much qualitative data is ordered (that is, where some characteristic A is bigger|better|stronger|whatever than another, B). However there is considerable doubt in most cases whether these orderings translate to numbers that can be meaningfully manipulated with statistical techniques.

More importantly though, we believe that the problems that have been identified in management research – not just the particular issues in the examples presented here, but the more general problem of practitioners seeing the research as irrelevant – should serve as a warning to researchers in the MABS community. We are a community that employs a wide range of tools and techniques, often 'borrowing' them in part or whole from other fields. This in itself is not a bad thing – indeed we may well discover new insights from such approaches – but *we must be aware of the assumptions and limitations of such approaches.* If we are to make a contribution to social science, and encourage researchers and practitioners from that discipline to work with us, we must be able to demonstrate not just that we can create simulations, but that we can produce results that are useful, meaningful and convincing.

References

1. Sichman, J.S.: Multi-agent-based simulation (MABS) – the international workshop series, website. http://www.pcs.usp.br/~mabs/ (2005)
2. Gilbert, N., Conte, R., Sichman, J.S.: ICMAS 98 workshop on multi-agent systems and agent-based simulation (MABS) website. http://cress.soc.surrey.ac.uk/mabs98.html (1998)
3. de Jong, A., de Ruyter, K., Wetzels, M.: Antecedents and consequences of group potency: A study of self-managing service teams. Management Science **51** (2005) 1610–1625
4. Hales, D.: Change your tags fast! a necessary condition for cooperation? In Davidsson, P., Logan, B., Takadama, K., eds.: Multi-Agent and Multi-Agent-Based Simulation: Joint Workshop MABS 2004, New York, NY, USA, July 19, 2004, Revised Selected Papers. Number 3415 in LNAI, Springer (2005) 89–98
5. Laursen, K., Salter, A.: Open for innovation: the role of openness in explaining innovation performance among U.K. manufacturing firms. Strategic Management Journal **27** (2006) 131–150
6. Sosa, R., Gero, J.S.: Social change: Exploring design influence. In Hales, D., Edmonds, B., Norling, E., Rouchier, J., eds.: Multi-Agent-Based Simulation III 4th International Workshop, MABS 2003, Melbourne, Australia, July 14th, 2003, Revised Papers. Volume 2927 of LNAI., Springer (2003) 106–119
7. Pfeffer, J., Fong, C.T.: The end of business schools? less success than meets the eye. Academy of Management Learning & Education **1** (2002)
8. Ghoshal, S.: Bad management theories are destroying good management practices. Academy of Management Learning & Education **4** (2005) 75–91
9. Bennis, W.G., O'Toole, J.: How business schools lost their way. Harvard Business Review (2005)
10. Antunes, L., Balsa, J., Urbano, P., Moniz, L., Palma, C.R.: Tax compliance in a simulated heterogeneous multi-agent society. In Sichman, J.S., Antunes, L., eds.: Multi-Agent-Based Simulation VI: 6th International Workshop, MABS 2005, Revised Papers. Volume 3891 of LNAI., Springer (2006)
11. Melo, A., Belchior, M., Furtado, V.: Analyzing police patrol routes with the simulation of the physical reorganization of agents. In Sichman, J.S., Antunes, L., eds.: Multi-Agent-Based Simulation VI: 6th International Workshop, MABS 2005, Revised Papers. Volume 2927 of LNAI., Springer (2006)
12. Rodrigues, M.R., Luck, M.: Analysing partner selection through exchange values. In Sichman, J.S., Antunes, L., eds.: Multi-Agent-Based Simulation VI: 6th International Workshop, MABS 2005, Revised Papers. Volume 3891 of LNAI., Springer (2006)
13. Sultanik, E.A., Peysakhov, M.D., Regli, W.C.: Agent transport simulation for dynamic peer-to-peer networks. In Sichman, J.S., Antunes, L., eds.: Multi-Agent-Based Simulation VI: 6th International Workshop, MABS 2005, Revised Papers. Volume 3891 of LNAI., Springer (2006)
14. Edmonds, B.: Against the inappropriate use of numerical representation in social simulation. Technical Report 04-131, Centre for Policy Modelling (2004) available at http://cfpm.org/cpmrep129.html.
15. Edmonds, B.: Assessing the safety of (numerical) representation in social simulation. In: Proceedings of the 3rd European Social Simulation Association conference (ESSA 2005), Koblenz, Germany (2005)

16. Academy of Management Review: Information for contributors. Academy of Management Review **30** (2005) p. 230
17. Academy of Management Journal: Information for contributors. Academy of Management Journal **48** (2005) p. 179
18. Strategic Management Journal website: http://www3.interscience.wiley.com/cgi-bin/jabout/2144/ProductInformation.html (2005)
19. Administrative Science Quarterly website: http://www.johnson.cornell.edu/publications/asq/contributors.html (2005)
20. Management Science website: http://mansci.pubs.informs.org/amission.html (2005)
21. Hackman, J., Wageman, R.: A theory of team coaching. Academy of Management Review **30** (2005) 269–287
22. Wageman, R.: How leaders foster self-managing team effectiveness: Design choices versus hands-on coaching. Organization Science **12** (2001) 559–577
23. Cronbach, L.: Coefficient alpha and the internal structure of tests. Psychometrika **16** (1951) 297–333
24. Cohen, S.G., Ledford, Jr., G.E., Spreitzer, G.M.: A predictive model of self-managing work team effectiveness. Human Relations **49** (1996) 643–676
25. Milton, L., Westphal, J.: Identity confirmation networks and cooperation in work-groups. Academy of Management Journal **48** (2005) 191–212
26. Edwards, J.R.: The study of congruence in organizational behavior research: Critique and a proposed alternative. Organizational Behavior and Human Decision Processes **58** (1994) 51–100
27. Edwards, J.R.: Alternative to difference scores: Polynomial regression analysis and response surface methodology. In Drasgow, F., Schmitt, N., eds.: Measuring and Analysing Behavior in Organizations: Advances in Measurement and Data Analysis. Jossey-Bass (2002) 350–400
28. Cohen, P.R.: Heuristic Reasoning about Uncertainty: An Artificial Intelligence Approach. Pitman, London (1985)

PartNET++: Simulating Multiple Agent Partnerships Using Dependence Graphs

Júlio de Lima do Rego Monteiro and Jaime Simão Sichman

Intelligent Techniques Laboratory, University of São Paulo,
Av. Prof. Luciano Gualberto, 158 – trav. 3, 05508-900 São Paulo SP, Brazil
{julio.monteiro, jaime.sichman}@poli.usp.br

Abstract. The PartNET++ system is an experimental multi-agent-based simulation tool that uses a new model based on hyper-graphs for understanding partnership formation among heterogeneous agents. Based on a previous tool called PART-NET, by Conte and Pedone [2], the agents of the system have goals to achieve and actions that lead to these goals, which must be performed by other agents. When choosing preferred partners, the agents may have different strategies (utilitarians, subtantialists and misers). In PART-NET, partnerships were restricted to two agents, and the authors have shown that some social hypotheses were validated, i.e., that in heterogeneous societies, utilitarians should have the best net benefit, followed by substantialists, then my misers. By using the PartNet++ simulator in several new experiments, it is shown in this work that results that are valid for partnerships between two agents can be generalized for multiple agents.

1 Introduction

Multi-Agent-Based Simulation (MABS) is increasingly becoming an important instrument for the social scientist. More and more, these researchers rely on agent-based models to formulate and even validate social theories. The major obstacle for this kind of approach is the steep curve in learning programming languages and complex APIs. However, there has been a number of MABS simulation environments and tools [4][5][6] made available recently that eases the process of experimenting with computer simulation for artificial intelligence researchers and social scientists.

The main goal of this work is to develop one such tool, focused on studying the partnership formation among multiple heterogeneous agents. This tool is the next generation of PART-NET [2], a tool that did the same kind of simulation, but restricted to analyzing partnerships between pairs of agents instead of multiple agents. Thus a secondary goal of this simulator is to show that the theories proposed in [2] are also valid for multiple agents, and not only for pairs of agents.

In the next section, the main concepts of the PART-NET simulator will be explained, the agent architecture will be outlined, and the four experimental hypotheses proposed. Moreover, the problems found when dealing with multiple agents partnership formation (instead of pairs) will be presented.

In section 3, the proposed solution to this limitation will be presented. The PartNET++ simulator will be explained, along with each and every parameter that can be configured in its user interface, allowing to setup and track the simulation.

J.S. Sichman and L. Antunes (Eds.): MABS 2005, LNAI 3891, pp. 14 – 23, 2006.

Section 4 will present some of the results obtained in testing the proposed experimental hypotheses, showing that the results obtained provide evidence to sustain that they are still valid for multiple agent partnerships.

The conclusions and future work are presented in section 5, along with further uses for the tool.

2 PART-NET

The PART-NET system [2] was originally developed to study partnerships among pairs of agents that use heterogeneous strategies.

This simulation tool intended to validate a social theory about the importance of having different strategies in a society where agents are able to perform actions and have a number of individual goals that need to be fulfilled by partnerships with other agents.

2.1 Agent Architecture

Each agent in PART-NET is comprised of a set of goals, and a set of actions. Each goal has an associated importance, and each action has a cost to be performed, both cost and importance are subjective and comparable values. A certain goal can be achieved by accomplishing an action of the same type.

The system only handles reciprocal partnerships [1], where a pair of agents, with different goals, will need the other agent's action to achieve his goal.

Besides their goals and actions, each agent in PART-NET follows a strategy that dictates what kind of partnerships will be sought. There are three different strategies available:

Utilitarians, that tries to maximize the importance of the achieved goal while minimizing the cost of the action used;

Substantialists, that choose partnerships with most important goals, no matter what the cost is;

Misers, that seeks the partnerships with minimum cost, no matter the goal importance.

These three strategies cover most of the reasonable stereotypical choices an agent may have when choosing partnerships.

2.2 Experimental Hypotheses

In a strong economical perspective, substantialists are less rational than utilitarians. In a social environment, however, an agent would not always try to maximize the net benefit; sometimes there are more important goals that need to be carried out first.

With this assumption in mind, the following experimental hypotheses have been defined:

H1. In heterogeneous societies, utilitarians should have the best net benefit, followed by substantialists, then by misers;
H2. In homogeneous societies, substantialists get better net benefits as there are more objectives per agent in the society;

H3. In heterogeneous societies, substantialists end up superceding the utilitarians in terms of accumulated net benefit;
H4. In heterogeneous societies, the accumulated net benefit is greater than the mean of the accumulated benefits of similar homogeneous societies.

These four hypotheses have been validated right using the original PART-NET, running thousands of simulations, as shown in [2].

2.3 Dealing with Multiple Partners

The PART-NET agent architecture is not complex enough to support the important concept of **plans**. By using plans, it would be possible to express that in order to achieve a certain goal a sequence of actions must be executed, instead of only one. These actions could then be carried out by multiple agents, instead of only one.

When calculating partnerships between pairs of agents, complex formalisms can be avoided. In PART-NET, a list of possible reciprocal partnerships was built in the beginning of the simulation, and this list was sorted according to each agent's strategies.

However, when dealing with partnerships among multiple agents, the social dependence networks needs a more strict formalism to allow the representation of such partnerships.

3 PartNET++ System

The adopted formalism in developing PartNET++ was social dependence graphs, proposed in [3], as detailed below.

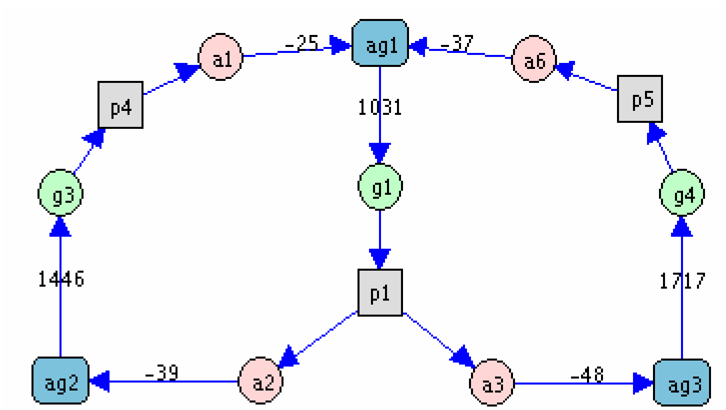


Fig. 1. Example of social dependence graph. In this figure, *ag1* has a goal *g1*, with importance 1031, there is a plan *p1* for this goal, composed of actions *a2* and *a3* that *ag1* cannot perform and can be fulfilled by agents *ag2* and *ag3*, costing respectively 39 and 48. Agent *ag2* has a goal *g3* with importance 1446, which can be achieved by a plan *p4* that can be completed by achieving action *a1*, which he cannot perform, but *ag1* can, at a cost of 25. Similarly, agent *ag3* desires the goal *g4* with importance 1717, using plan *p5*, composed only of action *a6* that costs 37 from *ag1*, thus a partnership can be formed between the three agents.

3.1 Dependence Graphs

In dependence graphs, all agents are interconnected together in a 4-partite graph, where nodes disposed in partitions for agents, goals, plans, actions and agents again, in this order.

The edges represent the structure of the society, having agents connected to goals using a positively weighted edge representing the importance of the goal and actions linked to the agents that can provide them over edges negatively weighted representing their cost. An example can be seen in Fig. 1.

3.2 PartNET++ Interface

The original PART-NET simulator was written in 1997, designed to run the necessary simulations to test the experimental hypotheses. All simulation configuration parameters and results were entered and shown via text files.

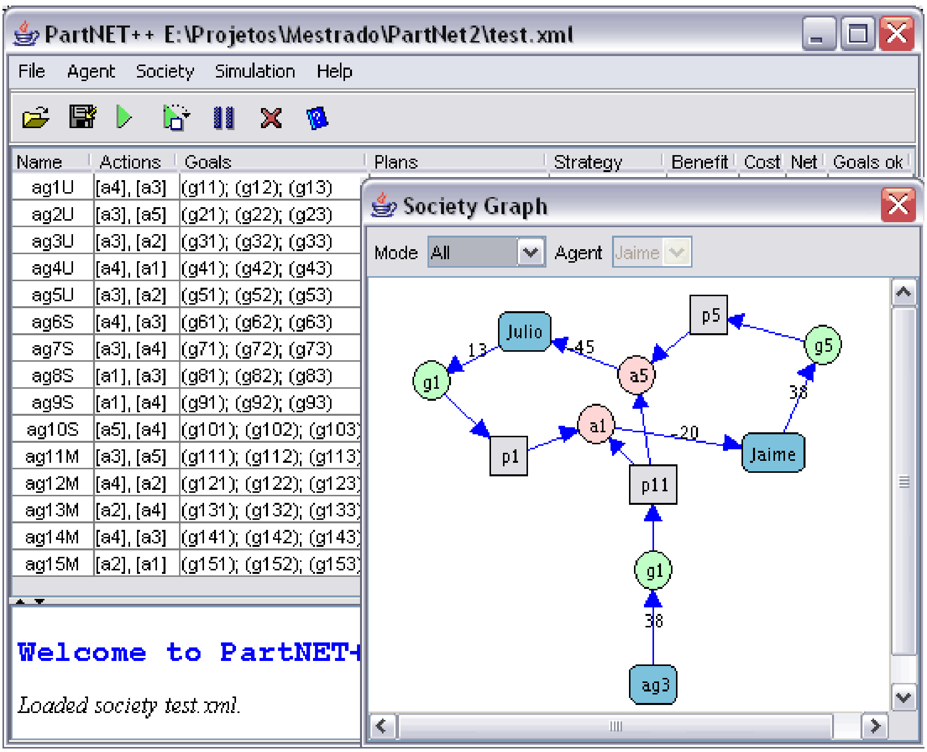


Fig. 2. The PartNET++ interface. In the upper part of the window, the main controls for the simulation can be seen, with the ability to load, save, start, step, pause and reset it. Below that, there is an agent summary showing all the information regarding the agents, updated as the simulation steps forward. As an overlay window, the user can track the social dependence graph, being able to visualize what dependence relations have been used in partnerships.

Hence, a secondary goal of PartNET++ was to implement a practical and easy to use graphical interface to configure, run and evaluate simulation results, with the ability to step through the simulation, step-by-step, examine the dependence graph and generate useful charts automatically (see Fig. 2).

Another interesting feature is the random society generator that can generate arbitrary societies based on customized parameters that will be explained below (see Fig. 3). This feature is crucial for mass simulation and hypothesis validation, allowing the researcher to build as many societies as needed for the tests.

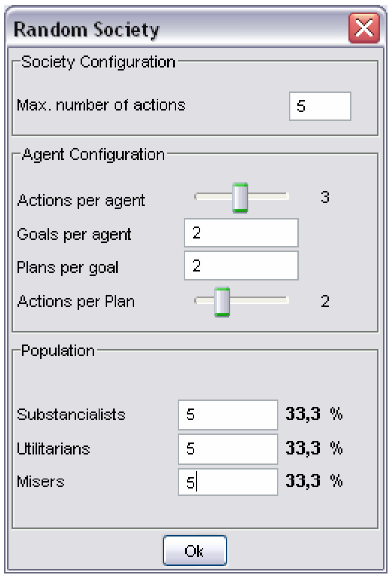


Fig. 3. Random society generator. For an explanation of each parameter, see section 3.3

3.3 Simulation Parameters

When setting up an experiment in PartNET++, it is needed to specify one or more societies. In general, the societies (i.e. agents, goals, plans and actions) are randomly generated for each run of the simulator, according to the specified parameters.

PartNET++ uses 10 different parameters to setup a simulation. The first six of them are common between PART-NET and PartNET++:

- **nU, nS, nM,** representing respectively the number of utilitarians, substantialists and misers in the society;
- **NAS,** representing the number of distinct actions available in the society;
- **APA,** that defines the maximum number of actions an agent can have;
- **GPA,** meaning the maximum number of goals that an agent can have.

The other four parameters are exclusive to the new simulator, because they are closely related to the plan concept and the dependence graph formalism:

- **PPG**, that measures how many plans can be generated for one goal belonging to an agent;
- **APP**, that dictates the maximum number of actions that can a plan be comprised of;
- **STR**, defining the social stratification level in this society (see below);
- **INT**, that measures the degree of social intolerance in the given society (see below).

Among the ten parameters, two of them deserve some special attention: *social stratification* (STR) and *social intolerance* (INT).

There are some cases in which an agent may search for an intermediary partner, since there is nothing he can offer to his partner directly, but he can do something for some other agent, that will, in his turn, pay the first agent's partner, allowing for a 3-agent partnership. This is calculated for each branch of the partnership, representing an action needed by a plan. Thus, the maximum number of agents that can participate as intermediaries in each branch of a partnership is called social stratification (STR). For partnerships that have branches with just the agent and his partner, the stratification is 1, like in the society detailed in Fig. 1, where there are 2 branches with STR=1, one between *ag1* and *ag2*, and the other between *ag1* and *ag3*.

On the other hand, an agent may find himself as an intermediary, willing to execute one of his actions for some other agent; however, there is no possible complete set of

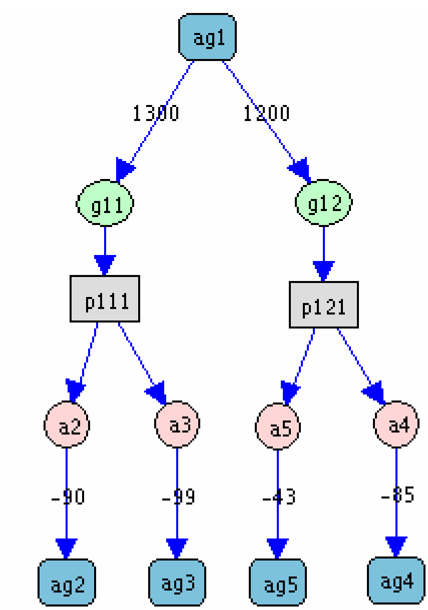


Fig. 4. Example of a dependence tree. In this figure, the primary agent is agent *ag1* that has two goals, *g11* and *g12*, with gains 1300 and 1200. For the goal *g1*, the only available plan is *p111* that is composed of actions *a2* and *a3*, that can be done by agents *ag2* and *ag3* at a respective cost of 90 and 99. The goal *g12* also can only be reached using plan *p121*, comprised of actions *a5* and *a4* that can be achieved by *ag5* and *ag4*, with the costs of 43 and 85 respectively.

actions that can be done in return that would accomplish any of his plans. If this agent is not intolerant, the partnership will happen, and he will get part of a plan accomplished, waiting to get the rest done in a future simulation cycle. The degree of social intolerance (INT) is then defined as the number of agents in a partnership that will have an intolerant behavior.

In PartNET++, for algorithm simplification, the INT parameter was always fixed at 1, meaning that only the primary agent in a partnership will require that all the actions in his plan be achieved, as explained below.

3.4 PartNET++ Algorithm

Each simulation is calculated in a series of steps. In every simulation step, each agent in a random order gets to choose partners for one of his goals, according to his strategy, evaluating the amount gained with the goal importance and what is spent with all the actions in the chosen plan. During this simulation step, this agent is called the *primary* agent, and all his possible partners are called *secondary* agents.

The first part of the algorithm assembles a dependence tree from the primary agent to all secondary ones (Fig. 4), then the primary agent chooses the most appropriate path in the tree, according to his strategy.

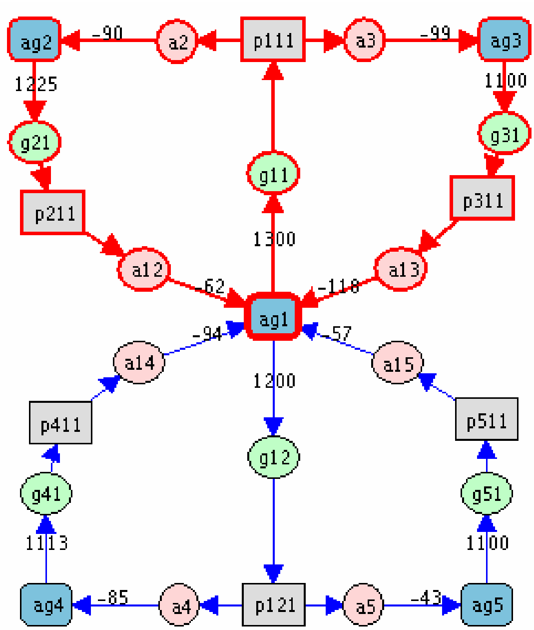


Fig. 5. Partnership between *ag1*, *ag2* and *ag3*. There are two branches for each of the two partnerships, the first two branches are among *ag1*, *ag2* and *ag3*, and the other two between *ag1*, *ag4* and *ag5*; supposing that *ag1* uses the utilitarian strategy he will choose the partnership with *ag2* and *ag3*, via plan *p111*, satisfying the goal *g11*, yielding a total net benefit of 1,111, instead of the 1,072 that would result from the partnership with agents *ag4* and *ag5*. The chosen partnership is highlighted.

This first part of the algorithm uses the notion of dependence networks exposed in [9], and is based on the algorithm used for the DEPINT system presented in [8].

From the utmost leaves of the dependence tree, that represent the chosen first order partners, the second part of the algorithm tries to find a path back to the primary agent, completing each partnership branch, always respecting the parameters STR and INT (see Fig. 5 for an example).

The process repeats itself until there are no more agents capable of accomplishing partnerships either by the lack of available actions or goals, or by the incompatibility of the remaining ones.

4 Results

Given a large set of experimental parameters for the PartNET++ simulator, one of the experimental hypothesis (H4) proposed before cannot be clearly confirmed using various experimental setups.

In general, to validate each of the hypotheses, each society in the experiment was run a hundred times, taking their averages and standard deviations into account.

In order to compare the multiple agents' partnerships results with those of pairs of agents, as studied in PARTNET, the parameters used in this experiment are compatible with the original PART-NET simulator. Therefore, the parameters APP, PPG, STR and INT are set to the value 1. Goal importance was randomly generated in the range of 0-1000. Action costs were also a positive integer from 0 to 1000 divided by the number of action in the associated plan, to keep the same scale.

In the following paragraphs, the experimental hypothesis H2 is validated. The data collected in the batches of simulations were organized in tables to facilitate the statistical calculation. The remaining hypotheses along with a more detailed description of the experiments can be found in [7].

4.1 Hypothesis H2

In order to validate hypothesis H2, that defines that substantialists gets better accumulated net benefits as there are more goals in the society, it is needed to setup two homogeneous societies with different quantities of goals per agent (GPA). The experimental setup is as follows:

#	nU	nS	nM	NAS	APA	GPA	PPG	APP	STR	INT
S1	-	60	-	20	5	10	1	1	1	1
S2	-	60	-	20	5	5	1	1	1	1

With different quantities of GPA, the net gains are naturally distinct as there are more ways for each agent to benefit. Therefore, to be able to compare these results appropriately, it is needed to divide each simulation totals by the number of partnerships, the result of this division is called partitioned benefit (BP).

The partitioned benefit for societies S1 and S2 are as follows:

$BP_1 = 2{,}223.7\ (\pm 46.4)$
$BP_2 = 2{,}071.3\ (\pm 50.7)$

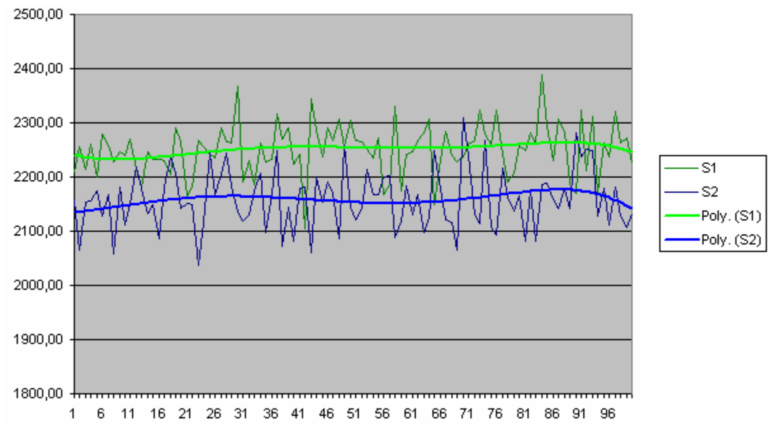


Fig. 6. Graphical results for hypothesis H2. Chart showing the partitioned benefit (BP) scored in each of the 100 runs of the simulation for societies S1 and S2. The graph includes a linear regression of the samples, to better recognize that S1 is significantly different from and has better scores than S2.

Using Student's T test for the two samples results a value of 22.17 that indicates the distribution of BP_1 is greater than BP_2 with an error of smaller than 0.0001%, as expected by hypothesis (see graphical results in Fig. 6).

5 Conclusions

Given the PartNET++ simulator was implemented in two steps. The first one, called PartNET+, was a remake of the original PART-NET, providing no added functionality, but consisting instead of an operational redesign, including a graphical user interface, the random society generator, chart generation and step-by-step simulation.

The second step, named PartNET++, was imbued with the task of implementing the new functionality to allow the new concept of plans and social dependence graphs.

Although there was no pre-made algorithm for finding cycles in 4-partite graphs, the solution achieved worked quite well, considering that the INT parameter was fixed to 1.

After the tool was completed, a set of more than 1400 simulations were executed to validate all the experimental hypotheses, which has required many hours of processing. Although there has been no mathematical analysis of the algorithm, every simulation was timed, resulting in the order of 12 seconds per run for societies of 60 agents with around 10 goals and actions, and 2 plans per goal.

According to the way it was built, the simulator can work in a limited scope of social problems, always involving partnerships between multiple agents with a relatively simple architecture. However, this tool can be extended to correspond to the needs of social scientists, interested in testing different or broader hypotheses.

The simulator can be quite didactic while explaining multiagent partnerships, as it includes a visual representation of the social dependence graph that evolves as each step of the simulation is run.

In the adopted model, every agent has the full knowledge of all dependency relations in the society. An interesting addition would be to limit the agent's perception, by using spatial constraints, or by creating a special kind of broker agents that know about the best partnerships.

Another interesting approach would be to include a reputation model for the agents, where each successful partnership would increase the agent's reputation with the partner.

The current implementation of the algorithm does not support the social intolerance parameter with values different than 1. In a future version of the algorithm, this limitation could be removed, in order to analyze its effect regarding the time complexity of the algorithm.

Finally, it would be nice to offer the user the possibility to propose new and different ad-hoc strategies, instead of always using the three original ones.

Acknowledgments

Jaime Simão Sichman is partially supported by CNPq, Brazil, grant numbers 304605/2004-2, 482019/2004-2 and 506881/2004-0.

References

1. Castelfranchi, C.; Conte, R.; Veneziano, V. The Computer Simulation of Partnership Formation. In: *Computational and Mathematical Organization Theory*, 4(4), p. 293-315, 1998.
2. Conte, R.; Pedone, R. Finding the Best Partner: The PART-NET System. In: *Proc. First International Workshop on Multi-Agent-Based Simulation* (MABS'98), Lecture Notes in Artificial Intelligence, Vol. 1534, p. 156-168. Springer-Verlag, Heidelberg, 1998.
3. Conte, R.; Sichman, J. Dependence Graphs: Dependence Within and Between Groups. In: *Computational and Mathematical Organization Theory*, 8(2), 87-112, 2002.
4. Gutknecht, O.; Ferber, J. The MADKIT Agent Platform Architecture. In: *Proc. 1st Workshop on Infrastructure for Multi-Agent Systems*, p. 48-55, Barcelona, Spain, June 2000.
5. Gasser, L.; Kakugawa, K. MACE3J: Fast Flexible Distributed Simulation of Large, Large-Grain Multi-Agent Systems. In: *First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2002
6. Langton, C. et al. The Swarm Simulation System: A Toolkit for Building Multi-agent Simulations. 1996, available at: http://www.santafe.edu/projects/swarm. Accessed in: May 15, 2004.
7. Monteiro, J. *Simulação de Parcerias Entre Agentes: Uma Extensão do Sistema PART-NET.* 143 p. Master Thesis, Escola Politécnica, Universidade de São Paulo , Brasil, 2004. (In Portuguese)
8. Sichman, J. *Du Raisonnement Social Chez les Agents: Une Approche Fondée sur la Théorie de la Dépendance.* 282 p. Doctorate Thesis. Institut National Polytechnique de Grenoble. Grenoble, France, 1995. (In French)
9. Sichman, J.; Conte, R.; Demazeau, Y.; Castelfranchi, C. A social reasoning mechanism based on dependence networks. In Tony Cohn, editor, *Proc. of the 11th. European Conference on Artificial Intelligence* (ECAI 1994), pg. 188-192. John Wiley & Sons Ltd., New York, 1994.

Analysing Partner Selection Through Exchange Values

Maira Ribeiro Rodrigues and Michael Luck

School of Electronics and Computer Science, University of Southampton, Southampton, UK
{mrm03r, mml}@ecs.soton.ac.uk

Abstract. Dynamic and resource-constrained environments raise interesting issues for partnership formation and multi-agent systems. In a scenario in which agents interact with each other to exchange services, if computational resources are limited, agents cannot always accept a request, and may take time to find available partners to delegate their needed services. Several approaches are available to solve this problem, which we explore through an experimental evaluation in this paper. In particular, we provide a computational implementation of Piaget's exchange-values theory, and compare its performance against alternatives.

1 Introduction

In many disciplines, like medicine and biology, researchers are discovering the advantages of collaborative research, in which different types of information and tools are exchanged in order to improve individual or global results [1, 2]. In an ideal situation, in which computational resources are plenty, individuals can collaborate with each other by performing services or giving access to information to any suitable (authorised) requester. However, when there are limited computational resources in the system, the participants must choose which requests to accept. Such collaborative systems are also dynamic, in the sense that participants change both their needs and provided services over time due, for example, to changes in research interests, and to the incorporation or development of new tools and information.

Dynamic and resource-constrained environments that take a collaborative approach to interaction raise interesting issues for partnership formation. First, because computational resources are limited, participants cannot always accept a request and may take time to find available partners to delegate their needed services. As a consequence, participants need some criteria to guide their decisions over interactions; more precisely, they need to choose which requests to accept when providing a service, and where to send requests that have a higher chance of being accepted when delegating a service. Second, because participants may change their needed and provided services over time, maintaining a collaborative interaction becomes more complicated, since the link between clients and services is not permanent. Third, if we assume that reciprocity is the basic motivation for interactions in collaborative environments, since it gives expectations of future interactions, this relation needs to be taken into account in the decision process over partners.

Guidance on partner selection in this context involves primarily two things: first, some criteria for choosing those participants that are more likely to perform a needed service, to avoid losing time trying to find an available interaction partner and, second,

J.S. Sichman and L. Antunes (Eds.): MABS 2005, LNAI 3891, pp. 24–40, 2006.

a strategy for accepting or refusing requests from other participants that favours the chances of future interactions (by considering the existence of relations of reciprocity and collaboration).

One way to select between interaction partners is to restrict the collaboration to situations in which a service dependency is observed. In this case, the concept of dependence [3, 4] can be used to motivate the exchange of services between participants and to influence their decisions about interactions. Although this dependence approach is powerful, we believe it is more suitable for situations in which the link between service provision and service request is static. Since dependence relations are built upon the services each individual needs and provides, if the relations change over time, the dependence network could lose accuracy, yet continual updates are likely to be expensive.

In response, we propose to address the problems of partnership formation and partner selection by using a system of exchange values, based on Piaget's theory [5]. Here, exchange values are viewed as the values that individuals associate with their interactions with other individuals, indicating the effort, cost, satisfaction, and benefit to each individual of the received or performed services and, thus, can influence the behaviour of interacting individuals. Piaget defines two main functions for exchange values in social interactions: as a means for individual decision-making, and as a regulation tool for guaranteeing the continuity of social interactions, since exchange values imply *moral commitments* made by individuals during their interactions (in the sense that they do not concern economic values, but instead what individuals owe to each other as a result of their interactions). Commitments are not used here as formal structures of mental representations of an obligation (as in [6]); rather, they are used to represent a moral obligation of one individual to return a favour to another, or to reciprocate an action received in the past.

The application of exchange values for supporting the exchange of services between agents was first proposed in [7], in which the social reasoning mechanism is based on Piaget's system. We believe this approach is suitable for addressing the problem of partner selection in dynamic and resource-constrained environments for two main reasons: first, exchange values provide a system of credits and debits that are seen as moral commitments between agents, which motivates and favours the chances of future interactions (for example, a credit that is gained by performing a service can be charged in the future) and, second, the set of exchange values of each individual (agent) is not related to the specific services they perform, but to the results of past interactions.

Other notions of values have appeared in the multi-agent systems literature. Miceli and Castelfranchi [8, 9], for instance, deal with the cognitive role of evaluations and values. They use values as a special kind of evaluation that mediates beliefs and goals in the agents' knowledge representation. Antunes and Coelho [10, 11] use a notion of "multiple values" to improve agents' decision-making process by explicitly including these values in their agent architecture, the BVG architecture. Although these approaches relate in one way or another to the concept of values, they neither encompass the idea of exchange, nor have the moral commitment implications (that suggest reciprocity and potentially regulation) of the notion of values in Piaget's approach.

In this paper, we investigate these approaches to interaction, through an implemented experimental testbed with a multi-agent simulation system. In particular, there has not been an implementation of the system of exchange values, so the results are novel and valuable. The key contribution is thus the computational model of the exchange values approach, and the associated experimental analysis of the performance of this approach in comparison to alternatives.

The paper begins with an introduction to the different social approaches we consider, followed by a description of the scenario we use, as well as the details of the strategies used (dependence-based and value-based) in a computational context. We end by presenting the experimental simulations and results obtained.

2 Background

This section describes two theoretical approaches that can be used to address the problem of partnership formation: dependency theory and Piaget's system of exchange values, both as a basis for decision-making about interaction partners.

2.1 Dependence Theory

The concepts of *social dependence* and *social power* in multi-agent systems were first proposed in [3, 4], with the basic idea that one of the fundamental notions of social interaction is the *dependence relations* between agents. This represents the situation in which an individual needs a resource (such as an object, a task to be performed, a piece of information) that he does not possess or have access to, but which can be provided by another individual, giving the latter influence over the former. The basic definition of social dependence is that an agent x depends on an agent y regarding an action a needed for achieving a goal g, if x is not capable of performing a but y is. In this case, agent y's action a is viewed by agent x as a resource for achieving g.

Associated with this is the notion of *power of influence* of an agent over another. Dependence relations can be associated with power of influence in the sense that when agent x depends on agent y for achieving a goal, it becomes susceptible to the influence of the latter agent. In this view, agents structure their interactions to take advantage of situations in which they can exert influence to ensure the compliance of agents with their requests. Consequently, the formalisation of these relations allows agents to achieve better interactions since an agent is able to control its interactions according to its own interests, by obtaining power through dependence relations.

In this way, some models for cooperation based on dependence relations have been proposed [12, 13], in which agents choose partners by identifying dependencies with other agents in a society. Also, work has been done to integrate the notion of dependence and norms with BDI-like agent architectures [14], and to describe social dependence networks as part of a formal framework [15].

2.2 Piaget's Theory of Exchange Values

Piaget's theory of exchange values studies and formalises the dynamics of values in such a way that they can form an exchange system [5], specifically in the context of the exchange of *services* between individuals, i.e., actions that an individual performs *on behalf* of another individual. The values that arise from such exchanges, which result from the individual's evaluation of the exchanged service, are called *exchange values*. They are seen as *moral values*, concerning *moral debits* (the obligation to perform new services in return for services previously received) and *moral credits* (the right to demand the performance of new services in return for services previously given).

Piaget's theory assumes two conditions for the existence of an exchange value system: the individuals involved in an interaction must share a *common scale of values* to ensure the compatibility of their evaluations of performed and received actions (services); and there must be conservation of the exchange values in time, so credits and debits acquired in the past continue to be valid in the future. If these two conditions hold, a system of exchange values can be seen as a mechanism for regulating (or coordinating) the social exchange of services between agents, guaranteeing their continuity (and thus the continuity of the society).

Four types of exchange values are involved in the exchanges between two individuals:

- a *renouncement value*, representing the effort, or *investment* of a provider in performing a service on behalf of another individual;
- a *satisfaction value*, representing the satisfaction of a requester with a received service;
- an *acknowledgement value*, representing the acknowledgement by the requester of a moral obligation with the provider, which is considered a debit; and
- a *reward value*, representing the recognition the provider thinks it received from the receiver for its work done, which is considered a credit.

These values are determined by the individuals over time, as a result of their exchanges with others. The renouncement and satisfaction values are viewed as real values in the sense that they are explicitly expressed by individuals. The acknowledgement and reward values are viewed as virtual values, since they are an internalisation of the real values, and provide the notion of acquired debits and credits. (In the remainder of this paper we use debit value instead of acknowledgement value, and credit value instead of reward value.)

Exchanges between two individuals α and α' occur in two stages, whose basic forms are as follows: in *Stage I*, an individual, say α, performs an action on behalf of another, say α', acquiring some credit for that action; and, in *Stage II*, α charges his credit, asking α' to perform some action for him in return. The event sequence for Stages I and II of a basic social exchange between α and α' is shown in Figure 1.

Stage I consists of four steps, as shown on the left of Figure 1:

1. α performs a service on behalf of α' and associates with this action a renouncement value (r_α);
2. α' receives the performed service, and associates a satisfaction value ($s_{\alpha'}$) with the received service;
3. α' then acknowledges a debit with α through a debit value ($t_{\alpha'}$);
4. α feels valued with the recognition of α', and associates to this valuation a credit value (v_α).

At the end of Stage I, α' has acquired a debit ($t_{\alpha'}$) with α, and α has acquired a credit (v_α) with α'. Note that since these are from the individual perspective of each, they may

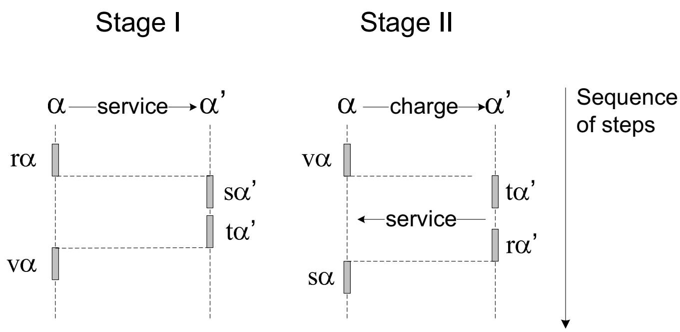


Fig. 1. The two stages of exchange

not be equal, and may not be known by the other. Later, α may charge the credit with α' by requesting performance of some service that benefits α in return. This begins Stage II of the exchange process, which follows the steps shown on the right of Figure 1, and is explained below:

1. α requests a service from α' in exchange for its credit v_α acquired in Stage I;
2. α' acknowledges its debit $t_{\alpha'}$ with α;
3. α' performs the requested service on behalf of α, and associates with this a renouncement value $r_{\alpha'}$;
4. α receives the service, and associates a satisfaction value s_α with this service.

When Stage II finishes, the exchange is said to be complete. In this way, Stage I represents an *accumulation* of exchange values, and Stage II represents their *realisation*, since debits and credits are first generated and then used in the exchange of concrete actions. If the amount assigned to each exchange value in one stage of the exchange is the same, i.e., if they directly correspond, the system is said to be in *equilibrium* with respect to that stage. The equilibrium situation for Stage I can be represented by the following logical implication:

$$\textbf{Implication} \qquad (r_\alpha = s_{\alpha'}) \wedge (s_{\alpha'} = t_{\alpha'}) \wedge (t_{\alpha'} = v_\alpha) \Rightarrow (v_\alpha = r_\alpha) \qquad (1)$$

From this, we conclude that if the amounts assigned to the exchange values resulting from the evaluations of the performed or received service are equivalent, then α is valued by α' proportionally to the service that was provided. The equilibrium situation for Stage II is represented in a similar way.

It is important to notice that all four values are related, because they all originate from the evaluation of the exchanged service. However, because this evaluation is subjective, and consequently individuals use different, personal criteria in the evaluation (for example, according to their character, personality, personal experience, etc.), these values need not correspond. In summary, perfect or fair judgement results in a direct correspondence between the exchange values and, consequently, in exchanges in equilibrium, while imperfect or unfair judgement results in a divergence of the exchange values and, consequently, there is no equilibrium.

3 Scenario

In an effort to experiment with these models, we have adopted the context of a dynamic collaborative research environment (as in Bioinformatics), in which researchers exchange information and tools in order to complete experiments or validate hypotheses. The environment is dynamic because participants are likely to change their needs and capacities over time due, for example, to changes in research interests, the appearance of new theories, or development of new tools. In this scenario, we assume participants are motivated to collaborate based on their expectations of reciprocated services. Also, their computational resources are limited, as every participant has a maximum number of services they can perform at one time. Therefore, when capacity is full, a participant cannot accept more requests to perform a service.

As a consequence, when a participant needs a service from others, she must choose those that are likely to perform the service and, when a participant receives requests from others, she must adopt a *criterion* for accepting or refusing those requests in a way that does not jeopardise her possibilities of future interactions. In this scenario, we assume that each researcher is represented by an agent that takes decisions about partners on her behalf.

The scenario thus consists of a group (or society) of agents, each of which has a set of services they need to execute, and a set a services they can provide (i.e., the agents are service clients and providers at the same time); agents exchange services in order to achieve individual goals. Since participants in a collaborative research environment are likely to change their needs over time, the agents that model them also change their set of needed services at a regular rate.

In addition, we assume that agents are not capable of performing their needed services (since we want to evaluate service exchange), but they always find interaction partners. High level relations between agents like goals or organisations are not considered, just relations between provided and needed services, so relationships like mutual or reciprocal dependence are omitted.

3.1 Strategies

A decision-making process towards the selection of partners, and the continuity of interactions, must establish not only the process of *choosing a partner*, but also the process of *analysing the received requests*, since the acceptance or refusal of a request will influence the chance of future interactions when reciprocal relations are considered. Therefore, an agent's decision over partners has two different contexts: when the agent receives a request and must decide whether to accept it; and when the agent must choose agents to send a service request, giving preference to those more likely to accept the request.

A common way to select between interaction partners is to restrict the collaboration to situations in which dependencies are observed. For example, an agent A is likely to collaborate with agent B by performing a service on its behalf, only if B is capable of performing a service that A needs. In the same way, if B needs to request a service from others, it will give preference to those with which it identifies a dependence (or that are seen as more likely to collaborate with it). In this case, the *criterion* used in the selection process is the *dependence*. The principle here is that the relationship between the services an agent needs and those services that others can provide determines if it is likely to be influenced by others, and can be used in the decision process. We call this approach the *dependence-based approach* for decision-making regarding interaction partners.

Another way of selecting between partners is to use exchange values as a basis for decision-making. This uses the basic exchange values system, according to which the agent's decision is not based on the services each agent can perform, but on the results of past interactions and expectations of future interactions. It implements a system of credits and debits, derived from the performed and received services, and is used to choose those agents that are more likely to perform tasks, as well as to choose whether to collaborate with other agents. In this case, the *criterion* used is the set of *exchange values*. The principle here is that the moral debits and credits acquired by an agent as a result of its interactions with other agents determine if it is likely to be influenced by others. We call this approach the *value-based approach* for decision-making. In what follows, we refer to agents using the former strategy, employing dependence-based reasoning, as *DAgents*, and the latter value-based agents as *VAgents*.

To analyse requests for a service, an agent considers: information on past interactions and on services provided and by whom, current capacity for service provision, and preferences regarding interactions. To analyse possible partners to send service requests, an agent considers information on past interactions, services provided, and preferences regarding interactions.

Information on past interactions and on services is used according to each approach. For example, the dependence-based approach uses information about service provision to infer the dependence between agents, while the value-based approach uses information on past interactions related to the history and state of exchange values.

In the context of this paper, we assume the requests are analysed as soon as they arrive, and that when the agent's current capacity is less than half full it will always accept requests. However, when the current capacity is equal to, or more than, half the maximum, the agent's decides whether to accept the request based on a set of preferences that are different for each approach. For example, the dependence-based approach has preferences related to dependence relations, while the value-based approach has preferences related to the state and history of exchange values.

In this scenario, agents request needed services according to the initial configuration, one after another, and must deal with any request messages that arrive. To decide to which agent a request is sent, and from which agents a request should be accepted, the agents take into account the limitations imposed by the environment, and also the criteria defined by each approach for partner selection, dependence-based or value-based.

3.2 Dependence-Based Agents

DAgents use information on dependencies to choose between interaction partners based on the assumption that partners with which a dependence is identified are considered to be more likely to accept requests. Conversely, requesters in a dependence relation can influence the decision when analysing incoming service requests. In order to derive information on dependencies, DAgents keep a service log with a record of the agents for which they performed a service.

Table 1. Dependence-based algorithm for sending service requests

01 For each sn_i in S_n do
02 search the resulting L_{pp} for sn_i;
03 *order L_{pp} giving priority to agents in L_{DepOn};*
04 send request for first agent in L_{pp};
05 if request refused then
06 repeat
07 send request to next agent in L_{pp} ;
08 until request accepted OR $L_{pp} = \phi$
09 if $L_{pp} = \phi$ AND request not accepted then
10 go to line 4 and repeat the process.

Table 2. Dependence-based algorithm for analysing incoming requests

01 For each sr_i do
02 if $(current\_capacity < maximum\_capacity/2)$ then
03 accept request;
04 if $(current\_capacity = maximum\_capacity)$ then
05 refuse request;
06 if $(current\_capacity >= maximum\_capacity/2)$ then
07 if requesting agent is in L_{IsDep} then
08 **accept** request;
09 else
10 **refuse** request.

Formally, if we take α' to be a dependence-based agent, L_{pp} the set of possible partners for α' when requesting a service sn_i from the needed services set S_n, sr_i a service being requested from α', L_{DepOn} the agents that depend on α' to perform a service, and L_{IsDep} the set of agents on which α' is dependent, we can define two algorithms to represent α's decision-making. The algorithm in Table 1 describes the decision process for sending requests for services, and the algorithm in Table 2 describes the decision process for analysing incoming requests.

In this approach, preference is given to agents with which a dependence relation is observed, since they are more likely to accept a request. In order to determine the current dependence relation, DAgents search the service log for agents for which they have previously performed a service (so that these agents depend on them for that service). When resources are limited (in that the current capacity is almost full), requests cannot always be accepted, and we assume that DAgents prefer to collaborate only with agents they depend on for performing a service, as shown in Table 2.

3.3 Value-Based Agents

Agents using the exchange values approach store four types of values [7]: renouncement (r), satisfaction (s), debit (t) and credit (v). For each interaction in which a Vagent α participates, it stores a current set of exchange values $V_{\alpha\alpha'}$, which contains the accumulated exchange values associated with the other agent involved (α'). VAgents also maintain an individual *history of interactions* in which they participate, with the resulting exchange values, and *exchange values state*, which is the set of exchange values (r,s,t,v) accumulated through interactions with other agents. After each interaction, VAgents calculate their exchange values based on the evaluation of the provided or received service, and update both their history of interactions and their exchange values state with the current set of exchange values.

Gains and losses determine how exchange values are accumulated; that is, if each accumulated exchange value should decrease or increase. For example, when an agent α acquires a debit with another agent α', this should increase its accumulated debit with α'. Similarly, when α pays its debit, this should decrease its accumulated debit with α'.

For VAgents, we take a simplified value set for exchange values, namely integers (though other models are possible), and define each increase or decrease during a single step of an exchange process to be by an integral number of units, $+n$ or $-n$. We also simplify the way exchange values are calculated, by assuming that the values are always in equilibrium.

As before, if we take α to be a VAgent, L_{pp} to be the set of possible partners for α when requesting a service sn_i from the needed services set S_n, sr_i a service being requested from α, $L_{HasCred}$ the set of partner agents with which α has credits, and

Table 3. Value-based algorithm for sending requests for services

01 For each sn_i in S_n do
02 search the resulting L_{pp} for sn_i;
03 *order L_{pp} giving priority to agents in $L_{HasCred}$;*
04 send request for first agent in L_{pp};
05 if request refused then
06 repeat
07 if refusing agent is in $L_{HasCred}$ then
08 devalue refusing agent;
09 send request to next agent in L_{pp} ;
10 until request accepted OR $L_{pp} = \phi$
11 if $L_{pp} = \phi$ AND request not accepted then
12 go to line 4 and repeat the process.

Table 4. Value-based algorithm for analysing incoming requests

01 For each sr_i do
02 if $(current\_capacity < maximum\_capacity/2)$ then
03 accept request;
04 if $(current\_capacity = maximum\_capacity)$ then
05 refuse request;
06 if $(current\_capacity >= maximum\_capacity/2)$ then
07 if sender is charging a credit then
08 **accept** request;
09 else if sender has negative exchange history then
10 **refuse** request;
11 else
12 **accept** request.

L_{HasDeb} the set of agents with which α has debits, we can specify two algorithms to represent the decision-making of α. In Table 3, we show how to send service requests and in Table 4, we show how to analyse incoming requests.

In this approach, preference is given to agents with which an agent has credits, since they are more likely to accept a request because they have a *commitment* with the requester, and are motivated to collaborate and pay their debits (a service was received, and a service is provided in return). Since the algorithm was defined in such a way that an agent first selects the possible partners for a needed service (that is, those that have the capability to provide it), and then uses the information about debits to order them, the situation in which an agent is not able to pay a debit because it is not able to perform the requested service is not considered.

If a request charging a credit is refused, it indicates that the other agent did not keep its commitment, so it is devalued by the requester (the refusing agent looses credits with the requesting agent). This indicates that the refusing agent is a poor partner for collaborative interactions. When resources are limited and requests cannot always be accepted, we assume that a VAgent prefers to collaborate with agents with which it has debits, since they will try to keep their commitments by paying their debits and completing the exchange, or with agents with a good collaboration history (and do not have a negative exchange result in the history of exchanges). In summary, collaboration is modelled by seeking complete exchanges, where services are received and performed in return.

4 Experiments

In seeking to determine which approach performs better in the changing environment, we have implemented an experimental testbed and undertaken some experiments to compare the success of each approach in exploring collaborative situations and finding interaction partners in dynamic, resource-constrained environments.

To do this, we measure the number of request messages that an agent must send until the service request is accepted (the *effort*), and the number of services an agent can delegate (the *task completion* or accomplishment). It is expected that agents with good performance will have a low value for effort (indicating that they were successful in choosing partners more likely to perform a service), and a high value for task completion (indicating that they were able to delegate more services). In summary, it is desirable that agents find partners quickly (and hence reduce the number of messages in the system), but also that the reasoning process does not consume too much time and consequently reduce the number of delegated services. Since we have restricted the scenario, assuming that there is always a provider for every available service, we guarantee that the task completion measure only represents tasks that could not be completed because partners did not accept them.

4.1 Simulation Configuration

A simulation in our experiments consists of a number of agents requesting services from, and performing services for, others at the same time, for a predetermined duration. For the agents' service configuration, each agent has a set of services it needs to request from other agents, and a set of services it can provide. The number of needed

Table 5. An example of service configuration

Agent	**Needs**	**Provides**
$Vagent1$	$S1, S2, S2, S1$	$S4, S3$
$Vagent2$	$S3, S4, S5, S3$	$S1, S2$
$Dagent3$	$S1, S3, S5, S4$	$S2, S6$
...		

and provided services is the same for all agents, but there is no relationship between the number of needed services and the number of provided services. In addition, all available services are allocated to at least one provider, and no agent can provide and request the same service. This kind of *service configuration* is shown in Table 5 where, for example, Vagent1 needs services $S1, S2, S2, S1$ in order, and can provide $S4$ and $S3$ (in any order, and possibly concurrently). The service configuration is changed by randomly replacing half the services in the set of needed services of all agents in the system. Each simulation has a basic simulation configuration including, for example, the duration of the simulation, total number of agents, total number of available services, maximum capacity for providing services at the same time, frequency of changes in service configuration, etc. We vary this basic configuration in different experiments in order to analyse the agents' behaviour under different conditions.

Also, to compare agent behaviour in uniform environments and heterogeneous environments, we defined some simulations to run with only one type of agent (single), and others with both types of agents (mixed). As a baseline, we also used a generic agent (GAgent) with random partner selection.

4.2 Results

First, we varied the number of changes in the service configuration, to see if there is a difference in the performance of each approach when there is no change at all in the environment, and when the changes are more frequent. For this experiment, we defined simulations, each with single groups of 30 VAgents, DAgents and GAgents.

The results, in Figures 2 and 3, show that VAgents generally performed better, in both effort and task completion, and also had stable behaviour in relation to the number of requests per service (effort) and delegated services (task completion), despite the changes. DAgents had an increase of effort and a decrease in the number of tasks completed in response to the increase in the number of changes in service configuration.

In the second experiment, we varied the number of agents in the system, to observe the behaviour of the agents when we scale up the system. Here, we also simulated single groups of VAgents, DAgents and GAgents and used a frequency of 25 changes per simulation.

The results in Figures 4 and 5 again show that VAgents perform better with a convergence to stable behaviour in terms of effort (in requests) but with a decrease in task completion as the number of agents increases. However, we observe that all approaches showed a decrease in their number of completed tasks in response to the increase in the number of agents. Interestingly, for this experiment we also measured the number of requests sent by the agents, to investigate if the decrease in task completion was due to an

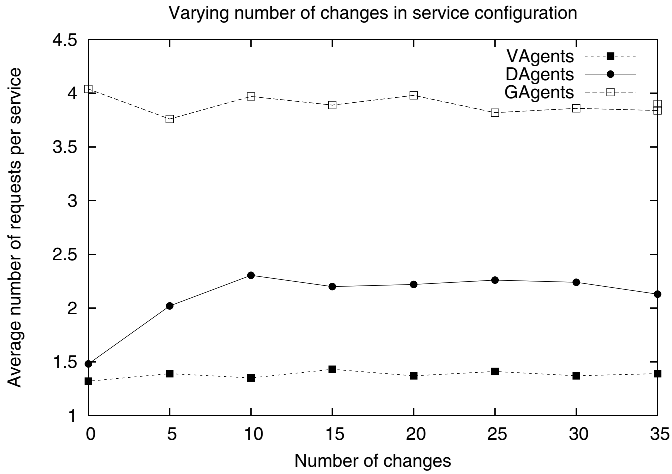


Fig. 2. Results for effort when varying frequency of changes in service configuration

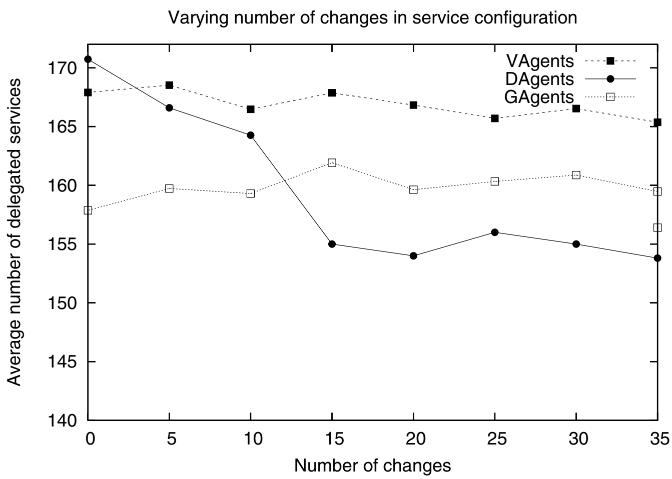


Fig. 3. Results for task completion when varying frequency of changes in service configuration

increase in the number of requests that could be responsible for making agents busier. We found that for GAgents, the number of request messages sent increased, but for the DAgents and VAgents the number decreased. This indicates that the latter spend more time in reasoning than in sending messages but, since the difference between task completion of VAgents and GAgents was relatively close, and the effort for VAgents was much better, we can conclude that as the number of agents increases the lower effort of VAgents compensates for the reasoning time. In comparison with DAgents, VAgents also performed better, stabilising at lower effort level and diverging from DAgents with better task completion as the number of agents increase.

Finally, we mixed the three types of agents in the same simulation, using the same number of agents of each type, keeping the parameters fixed to observe if their effort

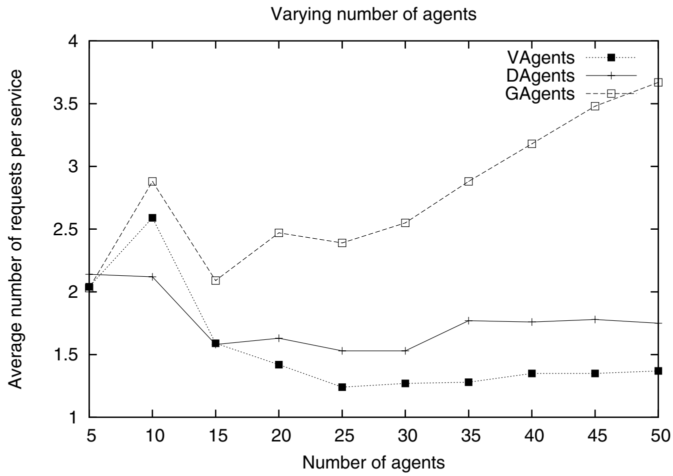


Fig. 4. Results for effort when varying the number of agents in the system

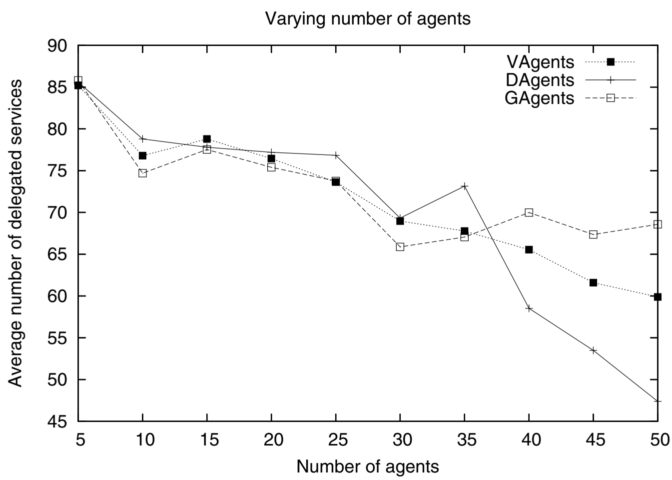


Fig. 5. Results for task completion when varying the number of agents in the system

stabilises over time, indicating that they can adapt to the heterogeneous environment. In this experiment, we mixed VAgents, DAgents and GAgents, 16 of each type, and used a frequency of 25 changes per simulation. The results for DAgents and VAgents in Figure 6 and 7 show that although DAgents do not have very high values for effort, they cannot achieve stable behaviour in effort. By contrast, VAgents have slightly higher values for effort at the start of the simulation, but achieved fairly stable behaviour towards the end, with very small variations in the number of requests per service.

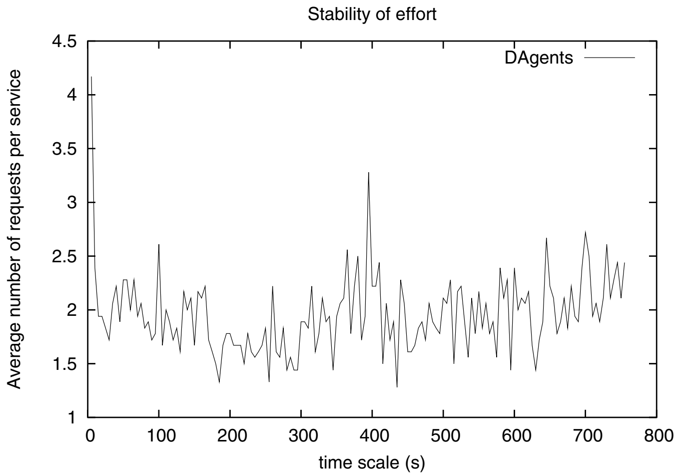


Fig. 6. DAgents stability of effort during the simulation

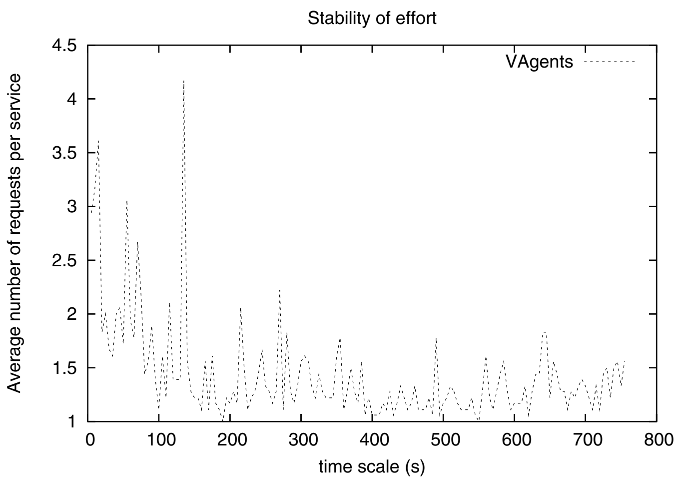


Fig. 7. VAgents stability of effort during the simulation

4.3 Discussion

We investigated three aspects of the partner selection approaches: the capacity to cope with changes in service configuration (in the first experiment), the capacity to cope with system scalability (in the second experiment), and the capacity to adapt to heterogeneous environments with agents using different strategies (in the third experiment). We also investigated the difference between the effort in finding interaction partners of agents using a strategy for partner selection as opposed to those not using one (by using random selection).

The results in the previous section show that in most cases VAgents perform better in the changing environment. However, we note that initial configurations can distort this as, for example, shown on Figure 3.

In the results for the second experiment (in Figures 4 and 5), the peaks we observe in the graphs are due to the distribution of services. Since each new agent added to the system has a different set of provided and needed services, a biased combination can either cause an agent to be busy if the new agents request services that were already scarce in the environment, or cause a more equal distribution of provided services if the new agents provide services that were scarce in the previous environment. Even though different service distributions could give different graphs, the results are not influenced, since the same configuration is used for all types of agents, and it is the relative comparison between them that identifies which approach performs better.

In summary, the results show that each approach has different advantages, and can therefore complement each other. For example, one possibility of combining both approaches is changing the algorithm for sending requests in a way that it first selects the agents with dependencies as possible partners, and then orders this set according to the exchange values approach, or vice-versa. The dependence approach could also be used in cases in which an agent has no credits or debits with the possible partners (as a result of not having encountered them previously).

5 Conclusions and Future Work

In this paper, we have described an experimental testbed with a multi-agent simulation system to simulate the process of interaction partner selection in dynamic and resource-constrained environments. To model agent decision-making in relation to interactions and interaction partners, we used two different approaches: a dependence-based approach and an exchange values-based approach. These address the issue of finding a partner more likely to accept requests, in contrast to alternative approaches that seek the partner with the highest utility (as proposed in [16, 17]).

We described the experiments undertaken to test the behaviour of agents using both approaches in the task of finding interaction partners, and specifically to analyse the behaviour of agents using exchange values in their reasoning, since there is no previous empirical evidence of this approach. The results showed that agents using exchange values-based decision-making were more stable in their behaviour, and needed less effort to find interaction partners than agents using a dependence-based approach, despite the changes in the environment. Also, by implementing a generic agent type that chooses interaction partners randomly, and comparing it to the other two types of agents, we observe that a strategy aimed at choosing interaction partners can significantly reduce the effort to find interaction partners, which can be very important when dealing with systems with resource limitations and dynamic behaviour.

Apart from these advantages, we believe that applying the system of exchange values in multi-agent systems can help more generally in regulating interactions, and motivating continuity of interactions, since exchange values imply moral and legal commitments, made by the agents during their interactions. Future work aims to analyse the combination of both approaches, dependence-based and exchange values-based (as proposed in [7, 18]) for the purpose of partnership formation. We also aim to extend agent reasoning and partner selection to include the notion of evaluation of provided and received services, so that the best interaction partner for delegating a service in terms of performance satisfaction can be considered.

Acknowledgements. The first author is supported by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) of the Brazilian Ministry of Education. We would also like to thank Antonio Carlos da Rocha Costa (UCPel, Brazil) for his valuable comments regarding the Piaget's theory of exchange values.

References

1. Stein, L.: Creating a bioinformatics nation. Nature **417** (2002) 119–120
2. Overbeek, R., Disz, T., Stevens, R.: The SEED: A peer-to-peer environment for genome annotation. Communications of the ACM **47** (2004) 47–50
3. Castelfranchi, C.: Social Power: A point missed in multi-agent, DAI and HCI. In Demazeau, Y., Müller, J.P., eds.: Decentralized A.I. Elsevier Science, Amsterdam (1990) 49–62
4. Castelfranchi, C., Miceli, M., Cesta, A.: Dependence Relations Among Autonomous Agents. In Werner, E., Demazeau, Y., eds.: Decentralized A.I. 3, Amsterdam, Elsevier Science (1992) 215–227
5. Piaget, J.: Sociological Studies. Routlege, London (1973)
6. Jennings, N.R.: Coordination techniques for distributed artificial intelligence. In O'Hare, G.M.P., Jennings, N.R., eds.: Foundations of Distributed Artificial Intelligence. (1996) 187–210
7. Rodrigues, M.R., da Rocha Costa, A.C., Bordini, R.H.: A system of exchange values to support social interactions in artificial societies. In: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems, Melbourne (2003) 81–88
8. Miceli, M., Castelfranchi, C.: A Cognitive Approach to Values. Journal for the theory of social behavior **2** (1989) 169–193
9. Miceli, M., Castelfranchi, C.: The Role of Evaluation in Cognition and Social Interaction. In Dautenhahn, K., ed.: Human Cognition and Social Agent Technology. Volume 19 of Advances in Consciousness Research. John Benjamins, Amsterdam (2000) 225–261
10. Antunes, L., Coelho, H.: Decisions based upon Multiple Value: The BVG agent architecture. In Barahona, P., Alferes, J.J., eds.: Progress in Artificial Intelligence, Proceedings of the Ninth Portuguese Conference on Artificial Intelligence. Volume 1695 of Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin (1999) 297–311
11. Antunes, L., Faria, J., Coelho, H.: Improving choice mechanisms within the bvg architecture. In Castelfranchi, C., Lespérance, Y., eds.: Intelligent Agents VII: Proceedings of the Seventh International Workshop on Agent Theories Architectures and Languages. Volume 1986 of Lecture Notes in Computer Science. Springer-Verlag, Boston (2001) 290–304
12. Sichman, J.S., Conte, R., Castelfranchi, C., Demazeau, Y.: A social reasoning mechanism based on dependence networks. In Cohn, A.G., ed.: Proceedings of the Eleventh European Conference on Artificial Intelligence, John Wiley and Sons (1994) 188–192
13. David, N., Sichman, J.S., Coelho, H.: Agent-based simulation with coalitions in social reasoning. In Moss, S., Davidson, P., eds.: MultiAgent-Based Simulation. Volume 1979 of Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin (2001) 245–266
14. Lopez, F.L., Luck, M., d'Inverno, M.: A framework for norm-based inter-agent dependence. In: Proceedings of the Third Mexican International Conference on Computer Science. (2001) 31–40
15. d'Inverno, M., Luck, M.: A formal view of social dependence networks. In Zhang, Lukose, eds.: Distributed Artificial Intelligence Architecture and Modelling: Proceedings of the First Australian Workshop on Distributed Artificial Intelligence. Volume 1087 of Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin (1996) 115–129

16. Saha, S., Sen, S., Dutta, P.S.: Helping based on future expectations. In: Second International Joint Conference on Autonomous Agents and Multiagent Systems, Melbourne (2003) 289–296
17. Sarne, D., Kraus, S.: Time-variant distributed agent matching applications. In: Third International Joint Conference on Autonomous Agents and Multiagent Systems, New York New York (2004) 168–175
18. Rodrigues, M.R., da Rocha Costa, A.C.: Using qualitative exchange values to improve the modelling of social interactions. In Hales, D., Edmonds, B., Norling, E., Rouchier, J., eds.: Multi-Agent-Based Simulation III. Volume 2927 of Lecture Notes in Computer Science. Springer-Verlag, Berlin (2003) 57–72

Automatic Tuning of Agent-Based Models Using Genetic Algorithms

Benoît Calvez and Guillaume Hutzler

Universite d'Evry-Val d'Essonne/CNRS, LaMI, UMR 8042,
523, Place des terrasses de l'Agora 91000 Evry, France
{bcalvez, hutzler}@lami.univ-evry.fr

Abstract. When developing multi-agent systems (MAS) or models in the context of agent-based simulation (ABS), the tuning of the model constitutes a crucial step of the design process. Indeed, agent-based models are generally characterized by lots of parameters, which together determine the global dynamics of the system. Moreover, small changes made to a single parameter sometimes lead to a radical modification of the dynamics of the whole system. The development and the parameter setting of an agent-based model can thus become long and tedious if we have no accurate, automatic and systematic strategy to explore this parameter space.

That's the development of such a strategy that we work on suggesting the use of genetic algorithms. The idea is to capture in the fitness function the goal of the design process (efficiency for MAS that realize a given function, realism for agent-based models, etc.) and to make the model automatically evolve in that direction. However the use of genetic algorithms (GA) in the context of ABS brings specific difficulties that we develop in this article, explaining possible solutions and illustrating them on a simple and well-known model: the food-foraging by a colony of ants.

1 Introduction

Agent-based simulation (ABS) is interested in the modelling and the simulation of complex systems. Its aim is to reproduce the dynamics of real systems by modelling the entities as agents, whose behavior and interactions are defined. A first validation of such models is obtained by comparing the resulting dynamics, when the model is simulated, with that of the real system (measured thanks to experimental data). Similarly, Multi-Agent Systems (MAS) are designed so as to accomplish a given function in a collective and decentralized way. The validation of the system is thus given by the fact that the function is realized and that it is efficient.

In both cases, one of the crucial aspects of the design process lies in the tuning of the model. Indeed, this kind of model is generally characterized by lots of parameters which together determine the global dynamics of the system. The search space is thus gigantic. Moreover, the behavior of these complex systems

J.S. Sichman and L. Antunes (Eds.): MABS 2005, LNAI 3891, pp. 41–57, 2006.

is often chaotic: on the one hand small changes made to a single parameter sometimes lead to a radical modification of the dynamics of the whole system; on the other hand some emergent phenomena are only produced in very specific conditions and won't occur if these conditions are not met. The solution space can thus be very small. As a consequence, the development and the parameter setting of an agent-based model may become long and tedious if we have no accurate, automatic and systematic strategy to explore the parameter space.

The approach that we suggest is to consider the problem of the development and the validation of ABS or MAS models as an optimization problem. The validation can thus be reformulated as the identification of a parameter set that optimizes some function. The optimization function for ABS would be the distance between the artificial model that we simulate and the real system. The optimization function for MAS would be the efficiency in the realization of the function. Given the large dimensionality of the problem, optimization techniques such as Genetic Algorithms (GA) can then be used to explore the parameter space and find the best parameter set with respect to the optimization function.

However the use of genetic algorithms in this context is not so simple. The first reason lies in the choice of the fitness function: agent-based systems or simulations are dynamic and often characterized by emergent and transitory phenomena, which complicates the measure of the fitness function. What has to be measured and when, are questions that strongly influence the characteristics of the models obtained by the algorithm, thus the quality of the results. If the fitness function is not carefully chosen, the resulting models will be optimized for that specific fitness function, which may not correspond to the initial goal of the designer of the model. The second reason is that no mathematical model allows to anticipate the dynamics of an agent-based model without executing it. The computation of the fitness function then requires the execution of the multi-agent system or simulation (and even several executions to take the stochasticity of the system into account), which implies a high computational cost. It is thus necessary to develop strategies to accelerate the convergence of the algorithm.

In section two we present the problematics related to the parameter tuning of an agent-based simulation. Then in section three we present the general framework of genetic algorithms and show the difficulties that arise from the application of these techniques to agent-based simulation. In section four, we propose guidelines for the use of genetic algorithms with agent-based simulation and show how it applies to the example of ant-foraging, before concluding in section five.

2 Parameter Tuning

2.1 Parameters of Agent-Based Models

In the context of agent-based simulation, a model and the simulator with which it is executed include lots of parameters. These parameters can be of different natures. Some parameters are peculiar to the simulator: the discretization step for the modeling of time and space for instance can be a fixed feature of the simulator. As a consequence, these parameters can generally not be modified by the user. For this reason, we do not include this type of parameters in our parameter space. We only include the parameters that are specific to the model. Some of them can be extracted from the knowledge of the field (either experimental or theoretical) and can thus be associated to fixed values. Other parameters have to be kept variable, which can be for different reasons: on the one hand, the knowledge of the field is generally not exhaustive (which is the reason why we build a model and simulate it); on the other hand, this knowledge may not be directly compatible with the model. In this case, a common approach can be to try some values and simulate the model to see how it behaves globally. What we propose is to have a general approach to automate this process.

This problem appears to be quite common in the modelling and simulation of any kind of complex systems. This is the case especially for social simulation, as it was shown by Sallans et al. [1]. In the context of integrated markets model, they present a model with lots of parameters, whose design and validation requires specific techniques to select appropriate values for the parameters.

2.2 Objective

Depending on the motivation of the modeling work, the criteria used to explore the parameter space will also be different. This motivation may be to model and simulate a real system, but it can also be to study the discrete models that may produce a given emergent phenomenon. Finally, the motivation may be to propose models that perform best in the realization of a specific function.

In the first case, we want to check if the simulated model correctly grasps the behavior of the real system. The validation of the model will thus be to have a behavior identical to (as close to as possible) experimental knowledge. The search problem can be seen as the search of the parameter set that minimizes the distance between real and simulated data.

Having a similar behavior can also mean that specific emergent phenomena known to occur in a real system can be observed in the simulation. Emerging ant lines for example, will only occur if the chemical trails have specific properties, as we will see in next section. The emergence of this phenomenon will thus be associated to specific parameter values, and the search problem will consist in searching the different ranges of parameters where an emergent phenomenon is observable. In some cases, choosing slightly different values may lead to completely different results during the simulation, which complicates a manual exploration of the parameter space and justifies the development of automatic techniques.

2.3 Example

We will present the parameter setting of an agent-based model with the example of ant foraging (we use the multi-agent programmable modeling environment `NetLogo` [2] and its “Ants” model). Figure 1 shows this example. The principle is that each ant searches for food and brings it to the nest secreting a chemical on the way back. When other ants feel the chemical, they follow the chemical way

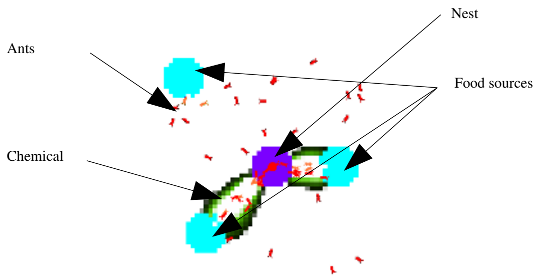


Fig. 1. Example of an ABS: Ant foraging

up to the food source, which reinforces the presence of the chemical and finally produces trails between the nest and the food sources. We can see ant lines emerging, which are similar to the ones that we can observe in natural conditions. In the model that we used, there is a nest in the center of the area of the simulation, and three food sources around the nest.

In this model, two parameters condition the formation of chemical trails. The first one is the diffusion rate of the chemical, which corresponds to the fact that a given proportion of the chemical will be diffused to the neighboring patches (regions of the environment) at the next time step. This is used to simulate the diffusion of the chemical in the atmosphere. The second parameter is the evaporation rate of the chemical, which corresponds to the fact that a given proportion of the chemical will disappear from the patch at the next time step. This is used to simulate the evaporation of the chemical in the atmosphere.

If we change the second parameter, we can get different behaviors of the simulation. Table 1 shows the variation in the evaporation rate.

Figure 2 shows the results during the simulation. We get two different global dynamics for the system. The first dynamics (simulation 1 and 3) is a random food search (either because there is too much chemical or because there is none). The second dynamics (simulation 2) is a food search with ant lines.

Table 1. Models with different evaporation rates

	Model 1	Model 2	Model 3
Diffusion rate	50	50	50
Evaporation rate	0	15	99

Fig. 2. Simulation results for the three models of table 1

Table 2. Models with slightly different parameters

	Model 1	Model 2	Model 3
Diffusion rate	40	50	60
Evaporation rate	15	15	20

Model 1 Model 2 Model 3

Fig. 3. Simulation results for the three models of table 2

For example, we can be interested more precisely in the dynamics of ant lines. Table 2 shows three models with small modifications for the two parameters. We can see in figure 3 that these small variations may lead to different dynamics: the difference lies in the way that food sources are exploited. In model 1, food sources are exploited in turn while in model 3, they are all exploited at the same time. As a result, we observe one, two or three ants lines.

2.4 Previous Work

Different methods have already been proposed to explore automatically the parameter space of discrete models. In the `NetLogo` platform for instance, the "BehaviorSpace" [2] tool allows to explore automatically and systematically the parameter space. This space is a Cartesian product of values that each parameter can take, some of them being chosen as fixed, others being limited to a subset of all possible values. However when we have lots of parameters, some of which can take a good many values (real-valued parameters for example), the parameter space becomes huge and the systematic exploration becomes impossible.

Other methods have been proposed, which differentially explore the whole parameter space, focusing on the most interesting areas. That's the case of the method developed by Brueckner and Parunak [3]. They use a "parameter sweep infrastructure", which is similar to the "BehaviorSpace" tool of `NetLogo` . However, to avoid a systematic exploration, they use searcher agents and introduce the fitness notion. The aim of a searcher agent is to travel in the parameter space to look for the highest fitness. Starting from a given location in the parameter space, searcher agents have two choices: move or simulate. Each agent chooses according to the confidence of the fitness estimate (proportional to the number of simulations at this point) and the value of the fitness. If it chooses to move, it heads for the neighboring region with highest fitness. A disadvantage of this method is that searcher agents may head for local fitness maxima.

Another method is to add knowledge to the agent-based model, as is the case with white box calibration [4]. The principle is to use the knowledge of the agent-based model to improve the tuning process. The aim is to reduce the parameter space by breaking down the model into smaller submodels, which can be done using different methods (General Model Decomposition, Functional Decomposition, ...). Each of the submodels is then calibrated, before merging them back to form the model. The division and fusion operations are the difficult steps of the method. The division operation, on the one hand, requires the addition of knowledge about the model, which may not be available. The fusion operation, on the other hand, has to merge calibrated submodels into a calibrated higher model, which is not automatic.

As we saw previously, Sallans et al. [1] have lot of parameters in their model. Some of these parameters are chosen based on initial trial simulations. The other parameters are chosen by the Metropolis algorithm, which is an adaptation of the Markov chain Monte Carlo sampling to do a directed random walk through parameter space. This method performs well on a continuous parameter space, but will hardly be usable when the parameter space is chaotic and there is the stochasticity in the simulation.

3 Use of Genetic Algorithms

As the tuning of the parameters of a model is a strongly combinatorial problem, we propose to use genetic algorithms, which generally provide good results on problems of this kind.

3.1 Principle of Genetic Algorithms

Genetic algorithms are a family of computational models inspired by evolution. They allow to solve various classes of problems, more specially optimization problems. Figure 4 shows the classic schema of genetic algorithms. In this framework, the potential solution of a problem is encoded on a linear data structure, which is called a chromosome. The algorithm works on a set of several chromosomes that is called a population. Operators are applied to this population.

The population of chromosomes is initialized randomly. Each chromosome is then evaluated using a fitness function, which measures how good this potential solution is with respect to the initial problem: it comes to give a score to each chromosome.

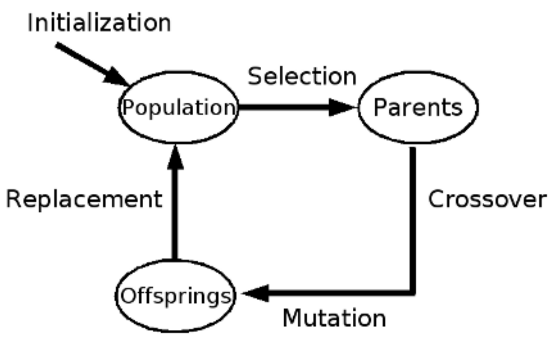


Fig. 4. Diagram of a standard genetic algorithm

A selection is made among the population of chromosomes: we obtain a new population named parent population. Recombination and mutation operators are then applied to this population: we obtain a new population named intermediate population. The recombination consists in swapping parts between two chromosomes. With this operation, we obtain two new chromosomes. That's the most frequent operator in genetic algorithms. Intuitively the role of this operator is to pick up the best part of chromosomes to obtain a better chromosome. The mutation consists in changing a part of a chromosome. This operation avoids converging prematurely to a local solution.

The new chromosomes of the intermediate population are evaluated. A new population is finally created from the initial population and the intermediate population, before starting again the whole process.

3.2 Choice of the Fitness Function

If we consider the exploration of the parameter space as an optimization problem, we need to define very carefully the function that will have to be maximized (or minimized) by the algorithm. This fitness function is of fundamental importance since the models that will be selected are the one that perform best with respect to this function. In the context of agent-based simulation, the choice of the fitness function is problematic for several reasons: as a first thing, it is not the result of a computation but the dynamics of a process that has to be assessed; secondly, emergent phenomena may be difficult to characterize quantitatively since they are often related to a subjective interpretation by a human observer.

Quantitative vs. qualitative. Validating an agent-based model by assessing the distance between the simulation and the real system can be done either quantitatively or qualitatively.

In the quantitative case, data are measured in the simulation and compared to data measured in similar conditions in the real system. The distance between the simulation and the real system is then the Euclidean distance between the two data vectors. If we try to select models that are optimized for the realization of some function, the fitness function can also be directly measured by the performance of the system for that function. In the case of ant foraging, this would correspond to the quantity of food retrieved to the nest after a given period or the time necessary to bring all the food back to the nest.

In the qualitative case, what is important is that a given emergent phenomenon be present in the simulation. The difficulty is then to translate this observation into a quantitative measure (the fitness function). In the example of ant foraging, we know from the observation of real ants that they organize dynamically along lines between the nest and food sources because of the creation of corresponding chemical trails. The fitness function could then be designed so as to reward models in which complete chemical trails are formed between the nest and the food sources. In some cases, the characterization of such emergent phenomena may not be so simple since it may be the result of a subjective interpretation by an observer, which cannot be captured easily by a quantitative measure.

A dynamic process. In classical optimization problem, the fitness function corresponds to the result of a computation. Therefore, the question of the time at which the measure should be made doesn't make sense: the measure is done when the computation has ended. On the contrary, agent-based simulations are dynamic processes that evolve along time and generally never end.

We can clearly see in the examples given in the previous section that the evaluation of the fitness function generally has to be done at a given time-step of the simulation. The choice of this time-step is not neutral and may greatly influence the performance of the genetic algorithm and the resulting model. If we try for example to select models that exhibit a given behavior that is transitory, it may not be sensible at the time-step chosen for the evaluation of the fitness function and the measure should thus be repeated at different time-steps.

Example. We can show with an example the difficulty of the choice of the fitness function. Figure 5 shows the foraging simulation at five different time-steps.

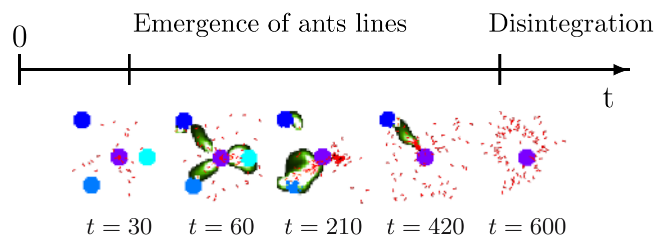


Fig. 5. Ant foraging at different time-steps

If we choose a quantitative fitness (quantity of food brought back to the nest after a given period) and if the computation of fitness function is too early ($t \leq 30$) or too late ($t \geq 600$), the value of the fitness will be the same regardless of the model: if the measure is too early, no food at all has been brought back, even in the best model; if the measure is too late, all the food has been brought back, even in the worst model. We have thus to choose the "right" moment to evaluate the fitness function (in this case between 60 and 420). However, this moment may not be the same for all different models, which suggest that it may be useful to evaluate the fitness at different time-steps.

If we choose a qualitative fitness (existence of ant lines) and if the computation of the fitness function is too early ($t \leq 30$) or too late ($t \geq 600$), we see no ant lines. Again, we have to choose the "right" moment to evaluate the fitness function (in this case between 60 and 420). But the number of ant lines also evolves during the simulation, which suggest again that the fitness should be evaluated at different time-steps on this interval.

3.3 Computation of the Fitness Function

Time. Since no mathematical model can anticipate the dynamics of an agent-based model without executing it, the computation of the fitness function requires one or even several simulations. This means that the time required to compute the fitness function will be significant (which is generally not the case for classical optimization problems). The global computation time to run the genetic algorithm is $(n \times N) \times T_f$ where n is the number of chromosomes, N is the number of generations, and T_f is the time to compute the fitness function. If T_f is 10 minutes, n is 20 chromosomes and N is 100 generations, then the global time to run the genetic algorithm is nearly two weeks. We must therefore find methods to reduce either the number of chromosomes, the time to converge towards an optimum or the time to compute the fitness function. We mainly studied the last possibility through distributed computation and fitness approximation[5].

Distributed computation. Since the different models are independent from each other, the evaluation of their fitness is also independent. Therefore each evaluation of the fitness (that is to say each agent-based simulation) can be done on a different computer. We can thus have several computers to simulate the models and use the master-slave parallel genetic algorithms [6], which improves the performance as compared to standard GA.

Fitness approximation. Fitness approximation comes to approximate the result of the simulation by a mathematical model, such as a neural network or a polynomial for instance. We tried this approach by training a neural network with test data. After the learning phase, we used it to compute the fitness, with the generation-based control approach, in which the whole population during η generations is evaluated with the real fitness function in every λ generations [7]. The results however were not so good and this approach has been temporarily abandoned. We suspect in that case that the approximation was not good enough to obtain satisfying results but this has to be explored in more details.

Stochasticity. Two agent-based simulations can generally bring slightly different results, even if the underlying model is exactly the same, due to the stochasticity of the model and of the simulator. One simulation is not enough to evaluate the fitness function: it can only be considered as an estimate for the fitness.

We studied the stochasticity of the "Ants" model and the `NetLogo` simulator. We chose three different fitness functions:

- the first fitness function is the quantity of food brought back between 100 and 200 simulation time-steps;
- the second fitness function is the time to bring back all the food to the nest;
- the third fitness function is the number of ant lines.

We assessed, depending on the number of simulations, the error rate in the estimation of the fitness as compared to the "real" fitness (estimated with 100 simulations). The results are shown in table 3.

We see, for instance, that a 5% error on the estimation of the fitness 2 can only be obtained by simulating the model more than five or ten times. But the stochasticity is more or less important depending on the fitness. Fitness 1 for example is much more sensitive than fitness 2.

Table 3. Example of the stochasticity

	Fitness 1	Fitness 2	Fitness 3
Number of Simulations	Error rate	Error rate	Error rate
1	$\simeq$34.57%	$\simeq$10.92%	$\simeq$13.28%
5	$\simeq$14.74%	$\simeq$4.85%	$\simeq$6.34%
10	$\simeq$10.3%	$\simeq$3.38%	$\simeq$4.39%
15	$\simeq$8.3%	$\simeq$2.85%	$\simeq$3.58%
20	$\simeq$7.26%	$\simeq$2.39%	$\simeq$3.15%
σ^*	0.41	0.14	0.17

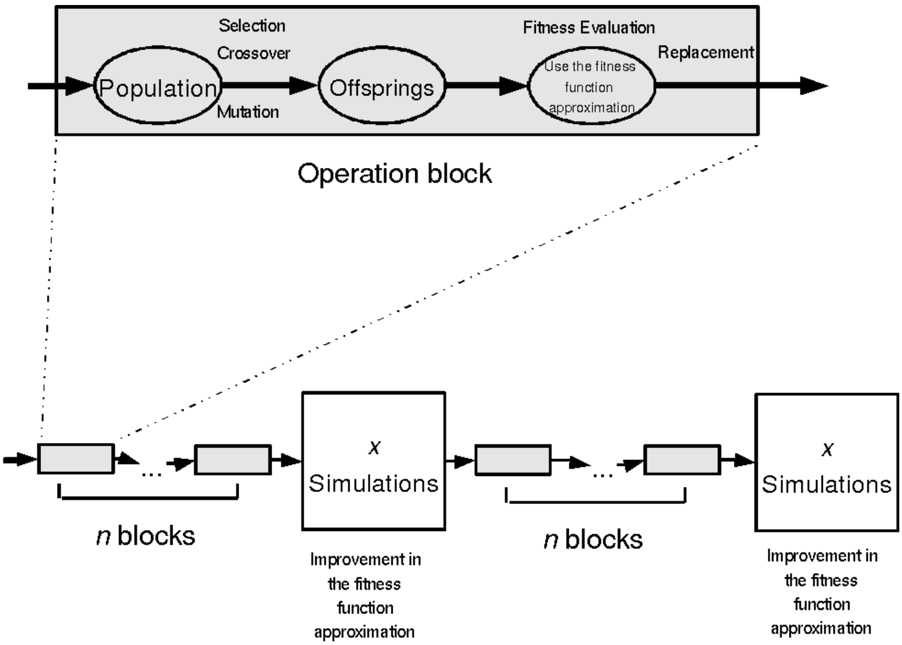


Fig. 6. Framework with the stochasticity

In such noisy environments, a first solution is to increase the size of the population [8, 9]. To multiply the number of the simulated models reduces the effect of the stochasticity. A second solution is to simulate each model several times to improve the evaluation of the fitness function. Both solutions greatly increase the number of simulations, thus the time, of the genetic algorithm.

Another solution is to use the same technique as with fitness approximation. A solution to the stochasticity problem is then to estimate the fitness of each model with one simulation, and each n generations of the GA (n to choose according to the stochasticity of the model and the desired quality of the estimation), to estimate the fitness of each model with x simulations. Figure 6 shows the general framework to take stochasticity into account.

We use the elitism genetic algorithm [10] that is to say we keep the best chromosomes during the algorithm, which allows to continuously improve the solution. Our implemented genetic algorithm replaces only 25 % of the population at each generation. Every 3 generations, we estimate the fitness of the models with more simulations.

4 General Framework and Application

Up to now, we identified several difficulties peculiar to agent-based systems for the application of genetic algorithms. We now propose a general framework or guidelines for the application of GA in this very specific context and show with several examples how it may be used.

1. determine the goal of the study;
2. elaborate the agent-based model;
3. choose the parameters of the model to evolve by genetic algorithms;
4. choose the fitness function : what do we want to optimize; when and how should we evaluate the function;
5. study the stochasticity of the model; simulate the model several times and study the results (calculate the standard deviation); determine the procedure for the exploration accordingly;
6. study the computation time of the simulation; If the simulation requires much time, use distributed computation; If the simulation requires too much time, use fitness approximation;
7. choose the number of chromosomes in the population;
8. run the genetic algorithm.

We will now detail how this may apply to the ant foraging example. In the three first examples, we use the model that has already been presented in the previous sections, with 10 ants. The main difference between the three first examples is related to the fitness function.

4.1 Example 1

Experience.

1. our goal is to optimize the foraging behavior;
2. the model is `NetLogo` "Ants" model;
3. the parameters that we evolve are the diffusion rate and the evaporation rate;
4. the fitness function is the quantity of food brought back between 100 and 200 simulation time-steps
5. The result of the study of the stochasticity is shown in table 3 for fitness 1; to evaluate the fitness function we use one simulation; every 3 generations we use 10 simulations to evaluate the fitness function;
6. the evaluation of the fitness function (that is to say a simulation) requires about 15 seconds; we use only one computer; one night of calculation is enough to compute 100 generations;
7. we take 20 chromosomes;
8. we run the genetic algorithms for 100 generations.

Results. We execute one run of the genetic algorithm. Figure 7 shows the results. The curve depicts the fitness of the best chromosome according to generations. And the crosses show the fitness of the best chromosome, computed by 10 simulations every 3 generations. The instability of this curve shows the stochasticity of the simulator and its model.

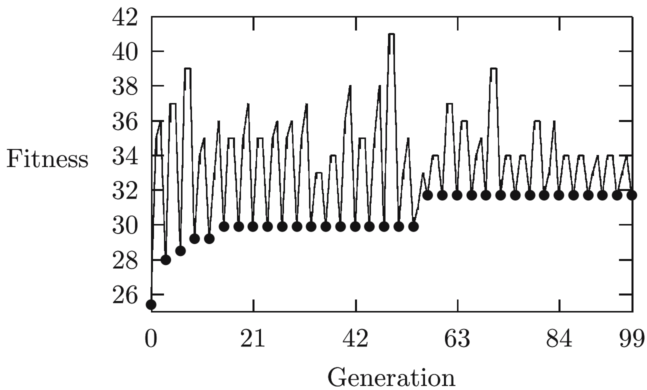


Fig. 7. Result of the genetic algorithm

At the end of the 100 simulation steps, the ant lines are built. During the next 100 simulation steps, the ants exploit the food sources using the chemical trails. The best models are the ones where the three food sources are exploited at the same time. However the evaporation rate is rather weak. (the evaporation rate is 8.1% and the diffusion rate is 88.6%) As a result, when there is no more food in a source, the chemical trail remains in the environment and the ants take time to exploit another source.

4.2 Example 2

Experience. The differences with example 1 are:

4. the fitness function is the time to bring all the food back to the nest;
5. the result of the study of the stochasticity is shown in table 3 for fitness 2;
6. the evaluation of the fitness function requires about 20 seconds for a good model and up to some minutes for a very bad model; we still use only one computer during one night.

Results. Figure 8 shows the results. Like the previous example, the best models are the ones where the three food sources are exploited at the same time. But the evaporation rates in this example are larger than in the previous example. It improves the dynamic behavior of ants: when a food source becomes exhausted, the ants quickly stop going to this source and search for other food sources.

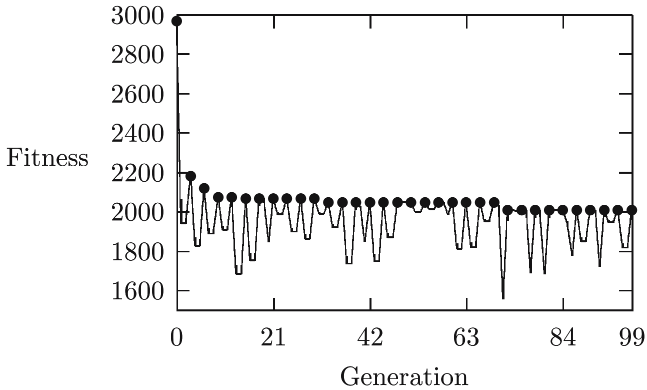


Fig. 8. Result of the genetic algorithm for example 2

4.3 Example 3 - Qualitative Fitness

Experience. The differences with example 1 are:

1. our goal is to obtain models in which ant lines can emerge;
4. the fitness is the number of ant lines; to simplify, we determine the number of continuous chemical trails between the nest and food sources; the number of trails is evaluated every 10 time-steps and the fitness is the sum of these values during 400 time-steps;
5. the result of the study of the stochasticity is shown in table 3 for fitness 3;
6. the evaluation of the fitness function requires about 30 seconds; we still use only one computer during one night.

Results. Figure 9 shows the results. The best models still exploit three food sources at the same time. Unlike the two previous examples, the evaporation and diffusion rates are but very small. This weakness allows to concentrate the chemical on narrow paths without covering all the environment, but it also reduces the flexibility of the behavior of the colony of ants.

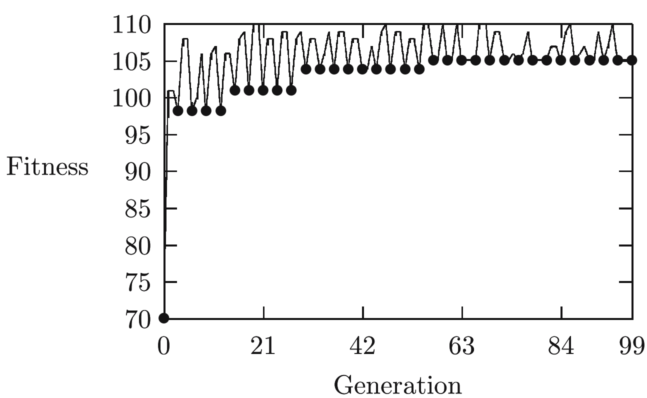


Fig. 9. Result of the genetic algorithm for example 3

4.4 Example 4

Experience. This example is different from the three previous examples. We try to tune the model parameters in relation with the experimental data. Firstly we generated data. To this end, we use the `NetLogo` "Ants" model, with the diffusion rate set to 50 and the evaporation rate set to 15. This model is considered as the initial model, which we simulate several times (1000 simulations) during 2500 simulation steps. Every simulation step, we record the quantity of food taken back to the nest. From the different simulations, we compute the average quantity of the food back to the nest at every step: we obtain a vector of the average food quantity of the initial model.

So, the differences with the example 2 are:

1. our goal is to obtain models as close to the data as possible;
2. the fitness function is the distance between the food quantity of the simulation and the vector of the average food quantity of the initial model.

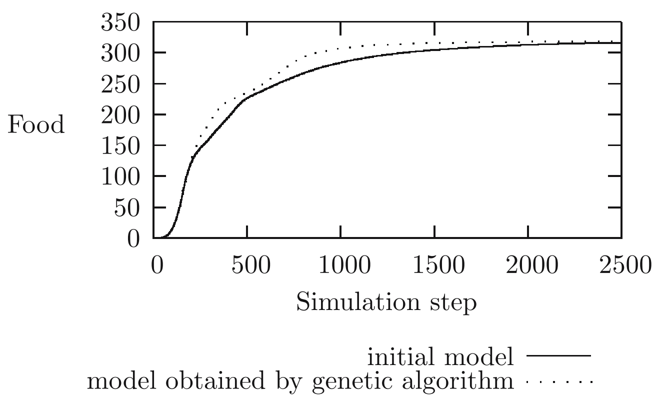


Fig. 10. Result of the genetic algorithm for example 4

Result. We execute several runs of the genetic algorithm. Figure 10 shows the results. The curves depicts the quantity of the food back to the nest every step of the best chromosome, and the average food quantity of the initial model. We can see that the two curves are very close, which indicates that the two models have a similar behavior. The parameter sets of these two models are but very different. The diffusion and evaporation rate of the model obtained by the genetic algorithm are respectively 10.86 and 20.33, and several executions of the genetic algorithm give very similar results. This indicate first that this example is not very sensible to the parameter setting, since different settings produce similar results. Why then does algorithm converges towards a specific setting? In this case, it appears that the algorithm converges towards parameter settings that are less sensitive to the stochasticity of the model. This may be explained by the fact that the fitness function is computed most of the time with only one execution of the simulation, thus favoring the models that are close to the reference model most of the time.

4.5 Example 5

Experience. We saw in the previous example that the reference model was not very sensitive to the parameter setting. In order to make it more sensitive, we increased in this example the size of the environment with a surface area multiplied by 4. We this increase the constraints imposed to the agents in the model.

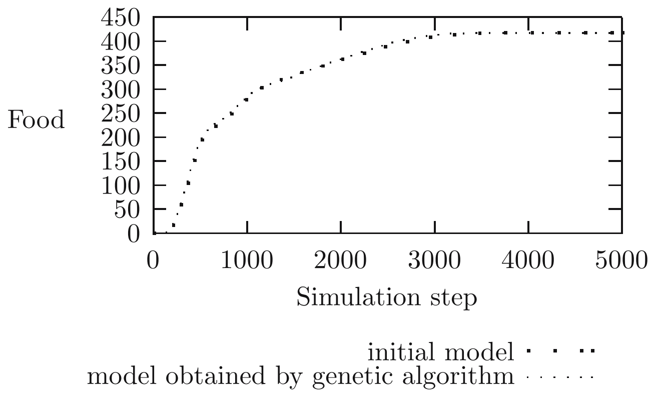


Fig. 11. Result of the genetic algorithm for example 5

Result. Figure 11 shows the results obtained with the genetic algorithm on this example. The curves depicts the quantity of the food back to the nest every step of the best chromosome, and the average food quantity of the initial model. Again, the two curves are very close. But this time, the parameter setting obtained by genetic algorithm is very similar to the parameter setting of the initial model.

4.6 Discussion

As we could already see with the study of the stochasticity, we obtain very different results depending on the choice of the fitness function. The models are strongly optimized for a specific fitness function and may not perform so well with another one. The optimization creates a loss of the flexibility of the dynamics of the agent-based model. A possible solution would be to use several different initial conditions to evaluate the fitness function. We could imagine in our example to vary the position and distance of the food sources, or to add new sources dynamically to select models that show high adaptation capabilities. This would however increase again the time necessary to run the algorithm.

The optimization by the genetic algorithm also depends on the constraints imposed to the agents in the model. If a model has lots of constraints (fewer resources for example), it is necessary that it optimizes its global functioning. On the contrary, if the resources are abundant, the pressure on the model to adapt and optimize its functioning will be weaker. As a result, the use of our approach will be mostly beneficial when constraints on the model are high. In the case of ant foraging, if the fitness is the time to bring all the food back, an important resource corresponds to the number of ants. The fewer the ants, the better the organization they will need to create so as to forage efficiently. On the contrary, if the ants are very numerous, the model will perform well whatever the chemical signalling. In some cases, it may thus be useful to strengthen the constraints on the model to obtain bigger improvements.

5 Conclusion

This paper presents a method, based on genetic algorithms, to explore automatically the parameter space of agent-based models. We explained the specific difficulties related to the use of this approach with agent-based models: the choice of the fitness function, the stochasticity, the computational cost. We then show some possible solutions. We finally suggest guidelines to help using genetic algorithms in this context.

Then we apply the method to some simple examples: the ant foraging with different fitness functions (both quantitative and qualitative). We obtain models that are optimized with respect to a given fitness function, which is chosen in relation with specific modeling goals.

The next step is to apply the method to a more complex example. We began a work for the simulation of the glycolysis and the phosphotranferase systems in Escherichia coli. In this work, we are interested in testing the hypothesis of hyperstructures [11]. The hyperstructures are dynamic molecular complexes, enzyme complexes in the case of this work. These complexes allow to improve the behavior of a cell : more flexibility, quicker adaptation. In our study, we have 25 kinds of molecules (or agents). There are altogether about 2200 agents in the simulation. We want to study the potential interest of hyperstructures for the cell. To do this we make the rates of enzymes association and dissociation variable. In this context, the simulation of a model lasts about 10 minutes, which imposes to use the methods described in this article like the distributed computation.

To explore this complex example, we will need to develop additional strategies to reduce the parameter space (e.g. by introducing coupling between parameters), to accelerate the evaluation of the fitness function (e.g. by developing approximation methods), and to accelerate the convergence of the algorithm (e.g. by using interactive evolutionary computation[12]). Finally, another important perspective is to explore the effect of varying dynamically the simulation conditions so as to produce more versatile models.

References

1. Sallans, B., Pfister, A., Karatzoglou, A., Dorffner, G.: Simulation and validation of an integrated markets model. J. Artificial Societies and Social Simulation **6**(4) (2003)
2. Tisue, S., Wilensky, U.: Netlogo: Design and Implementation of a Multi-Agent Modeling Environment. Proceedings of Agent 2004 (2004)

3. Brueckner, S., Parunak, H.V.D.: Resource-Aware Exploration of the Emergent Dynamics of Simulated Systems. AAMAS 2003 (2003) 781–788
4. Fehler, M., Klügl, F., Puppe, F.: Techniques for analysis and calibration of multi-agent simulations. In Gleizes, M.P., Omicini, A., Zambonelli, F., eds.: ESAW. Volume 3451 of Lecture Notes in Computer Science., Springer (2004) 305–321
5. Jin, Y.: A comprehensive survey of fitness approximation in evolutionary computation. Soft Computing Journal (2003)
6. Cant-Paz, E., Goldberg, D.E.: Efficient parallel genetic algorithms: theory and practice. Computer Methods in Applied Mechanics and Engineering **186** (2000)
7. Jin, Y., Olhofer, M., Sendhoff, B.: A Framework for Evolutionary Optimization with Approximate Fitness Functions. IEEE Transactions on Evolutionary Computation **6**(5) (2002) 481–494
8. Beyer, H.G.: Evolutionary Algorithms in Noisy Environments: Theoretical Issues and Guidelines for Practice. Computer methods in applied mechanics and engineering **186** (2000)
9. Goldberg, D.E., Deb, K., Clark, J.H.: Genetic algorithms, noise, and the sizing of populations. Complex System **6** (1992)
10. Beker, T., Hadany, L.: Noise and elitism in evolutionary computation. In: Soft Computing Systems - Design, Management and Applications. (2002) 193–203
11. Amar, P., Bernot, G., Norris, V.: Modelling and Simulation of Large Assemblies of Proteins. Proceedings of the Dieppe spring school on Modelling and simulation of biological processes in the context of genomics (2002) 36–42
12. Takagi, H.: Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. In: Proceedings of the IEEE. Volume 89., IEEE Press (2001) 1275–1296

Formal Interpretation and Analysis of Collective Intelligence as Individual Intelligence

Tibor Bosse and Jan Treur

Vrije Universiteit Amsterdam, Department of Artificial Intelligence,
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{tbosse, treur}@cs.vu.nl
http://www.cs.vu.nl/~{tbosse, treur}

Abstract. This paper addresses the question to what extent a process involving multiple agents that shows some form of collective intelligence can be interpreted as a single agent. The question is answered by formal analysis. It is shown for an example process how it can be conceptualised, formalised and simulated in two different manners: from a single agent (or cognitive) and from a multi-agent (or social) perspective. Moreover, it is shown how an ontological mapping can be formally defined between the two formalisations, and how this mapping can be extended to a mapping of dynamic properties. Thus it is shown how collective behaviour can be interpreted in a formal manner as single agent behaviour.

1 Introduction

Many processes in the world can be conceptualised using an agent metaphor, as a single agent (or cognitive) process or a multi-agent (or social) process. Especially for processes that are distributed, it is natural to describe them as a group of interacting agents. If a group of agents acts in a coherent way, however, one is often tempted to intuitively and informally interpret the process in singular form as a collective, and, in fact, as one individual (super)agent. The question addressed in this paper is whether in certain cases such an informal interpretation of a multi-agent system, acting in a collective manner, as an individual can be supported by a formal analysis. The approach to address this question is by formally defining an interpretation mapping between a conceptualisation of a process as a multi-agent system and a conceptualisation of the same process as an individual.

The prerequisites to undertake such a formal analysis concern formalisations of the notion of agent, single agent behaviour and multi-agent behaviour, and the notion of interpretation mapping. More specifically, what is needed is a formal notion of what an agent is in the sense of

- distinctions between the agent's internal mental processes, the agent's body, and the agent's environment
- interactions and relationships between mental aspects and body aspects
- interactions and relationships between agent and environment, including interactions with other agents

J.S. Sichman and L. Antunes (Eds.): MABS 2005, LNAI 3891, pp. 58 – 74, 2006.

Furthermore, formalisations of single agent behaviour and multi-agent behaviour are needed that cover

- the externally observable behaviour
- the underlying internal processes

Moreover, a formal notion of interpretation mapping of a single agent conceptualisation into a multi-agent conceptualisation is needed that

- maps ontological concepts describing a conceptualisation of a process from an individual perspective to ontological concepts describing a conceptualisation of the same a process from a multi-agent perspective
- covers mapping of individual mental state properties for the single agent conceptualisation to shared mental state properties for the multi-agent conceptualisation
- covers the mapping of dynamic aspects of single agent behaviour onto those of multi-agent behaviour

In this paper for these three notions formalisations are provided and used to indeed achieve a formalisation of how a collective can be formally interpreted as an individual.

The formalisation is evaluated for the case of collective behaviour of an ants colony. The intelligence shown by ant colonies are an interesting and currently often studied example of collective intelligence [1], [4], [8]. In this case by using pheromones the external world is exploited as a form of extended mind; cf. [2], [3], [6], [13], [14]. The analysis of this case study comprises on the one hand a multi-agent model, simulation based on identified local dynamic properties, and identification of dynamic properties for the overall process. On the other hand the same is done for an alternative model based on a single agent with internal mental states, and the two models are related to each other via the interpretation mapping.

In Section 2, a formalisation of basic agent concepts will be introduced. Section 3 explains, using a simple example, the idea of the basic formal ontology mapping between state properties in a single agent conceptualisation and state poperties in a multi-agent conceptualisation. In Section 4 this notion of basic interpretation mapping of state properties is applied to two conceptualisations of the more complex ant colony example, the central case study in the paper. Section 5 discusses the dynamics for the two conceptualisations of the ant colony example in more detail, which leads to formal specification of executable local dynamic properties that have been used for simulation. In Section 6 the basic interpretation mapping for state properties is extended to dynamic properties, thus obtaining an interpretation mapping between the two conceptualisations of the dynamics of the example ants colony process. Section 7 is a final discussion.

2 Basic Agent Concepts

The agent perspective entails a distinction between the following different types of ontologies:

- an ontology for *internal mental properties* of the agent A (MentOnt(A)),
- for properties of the agent's (physical) *body* (BodyOnt(A)),

- for properties of the (sensory or communication) *input* (InOnt(A))
- for properties of the (action or communication) *output* (OutOnt(A)) of the agent, and
- for properties of of the *external* world (ExtOnt(A)).

For example, the property 'the agent A feels pain' may belong to MentOnt(A), resp. BodyOnt(A), whereas 'it is raining' and 'the outside temperature is 7° C' may belong to ExtOnt(A). The agent input ontology InOnt defines state properties for received perception or communication, as an in-between step from environment or body state properties to internal mental state properties, the agent output ontology OutOnt defines state properties that indicate initiations of actions or communications of the agent, as an in-between step from internal mental state properties to environment or body state properties. The combination of InOnt and OutOnt is the *agent interaction ontology*, defined by InteractionOnt = InOnt $\cup$ OutOnt.

To formalise state property descriptions of the types introduced above, ontologies are specified in a (many-sorted) first order logical format: an ontology is specified as a finite set of sorts, constants within these sorts, and relations and functions over these sorts. The example properties mentioned above then can be defined by nullary predicates (or proposition symbols) such as itsraining, or by using n-nary predicates (with $n \geq 1$) like has_pain(A) and has_temperature(environment, 7).

For a given ontology Ont, the propositional language signature consisting of all *state ground atoms* based on Ont is denoted by APROP(Ont). The *state properties* based on a certain ontology Ont are formalised by the propositions that can be made, using (using conjunction, negation, disjunction, implication) from the ground atoms. The notion of state as used here is characterised on the basis of an ontology defining a set of physical and/or mental (state) properties that do or do not hold at a certain point in time. In other words, a *state* S is an indication of which atomic state properties are true and which are false, i.e., a mapping S: APROP(Ont) $\rightarrow$ {true, false}.

To describe the internal and externally observable dynamics of the agent, explicit reference is made to time. Dynamics will be described as evolution of states over time. Dynamic properties can be formulated that relate a state at one point in time to a state at another point in time. A simple example is the following informally stated dynamic property for belief creation based on observation:

'if the agent observes at t1 that it is raining, then the agent will believe that it is raining'.

To express such dynamic properties, and other, more sophisticated ones, the sorted predicate logic *Temporal Trace Language* (TTL) is used [10]. Here, a *trace* over an ontology Ont is a time-indexed sequence of states over Ont. TTL is built on atoms referring to, e.g., traces, time and state properties. For example, 'in trace γ at time t property p holds' is formalised by state(γ, t) |= p. Here |= is a predicate symbol in the language, usually used in infix notation, which is comparable to the Holds-predicate in situation calculus. Dynamic properties are expressed by temporal statements built using the usual logical connectives and quantification (for example, over traces, time and state properties). For example, the dynamic property put forward above can be expressed in a more structured semiformal manner as:

'in any trace γ, if at any point in time t1 the agent A observes that it is raining,
then there exists a time point t2 after t1 such that at t2 in the trace the agent A believes that it is raining'.

In formalised TTL form it looks as follows:

$$\forall\gamma\ \forall t1\ [\ state(\gamma, t1) \models observes(A, itsraining)\ \Rightarrow\ \exists t2 \geq t1\ \ state(\gamma, t2) \models belief(A, itsraining)\ ]$$

Based on TTL, a simpler temporal language has been defined to specify simulation models. This language (the *leads to* language) enables to model direct temporal dependencies between two state properties in successive states. This executable format is defined as follows. Let α and β be state properties of the form 'conjunction of atoms or negations of atoms', and e, f, g, h non-negative real numbers. In the *leads to* language $\alpha \bullet\!\!\rightarrow\!\!\rightarrow_{e, f, g, h} \beta$, means:

If *state property* α *holds for a certain time interval with duration g,*
then *after some delay (between e and f) state property* β *will hold for a certain time interval of length h.*

For a precise definition of the *leads to* format in terms of the language TTL, see [10]. A specification of dynamic properties in *leads to* format has as advantages that it is executable and that it can often easily be depicted graphically.

3 The Basic Interpretation Mapping

In this section it is discussed how a conceptualisation based on a single agent and individual (internal) mental state properties can formally be mapped onto a conceptualisation based on multiple agents and shared (for the sake of simplicity assumed external) mental state properties. Here this ontological mapping is only given in its basic form, for the state properties. In Section 6 the basic mapping is extended to temporal expressions describing behaviour.

First, consider Figure 1. This figure depicts a simple case of a single agent A with behaviour based on an *individual internal* mental state property m1. The solid arrows depict temporal *leads to* relationships. Mental state property m1 (temporally) depends on observations of three world state properties c1, c2, c3. Moreover, action a1 depends on m1.

Now consider Figure 2. This figure depicts a group of agents A1, A2, A3, A4 with behaviour based on a physical external world state property m2 that serves as a *shared external* mental state property.

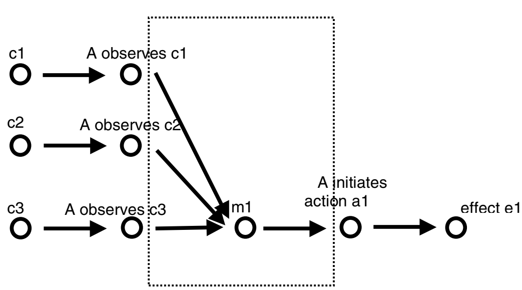


Fig. 1. Single Agent behaviour based on an internal mental state

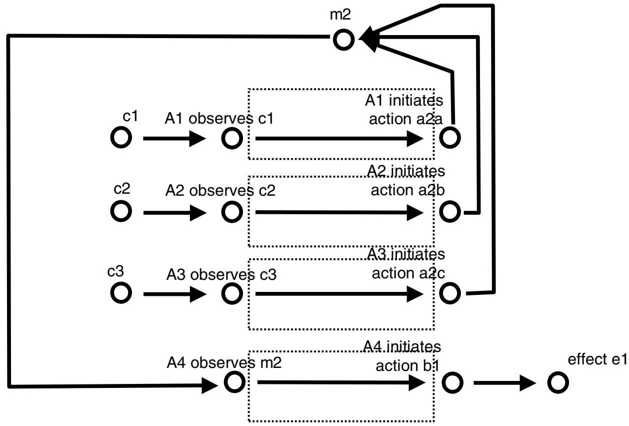


Fig. 2. Multi-Agent behaviour based on a shared external mental state

To create this shared mental state property, actions a2a, a2b, a2c of the agents A1, A2, A3 are needed, and to show the behaviour, first an observation of m2 by agent A4 is needed. Note that here the internal processing is chosen as simple as possible: stimulus response. The interaction between agent and external world is a bit more complex: compared to a single agent perspective with internal mental state m1, extra actions of some of the agents needed to create the external mental state property m2, and additional observations are needed to observe it.

To make the similarity between the two different cognitive processes more precise, the following mapping from the nodes (state properties) in Figure 1 onto nodes in Figure 2 can be made (see Figure 3):

External world state properties

φ: c1 → c1
φ: c2 → c2
φ: c3 → c3
φ: effect e1 → effect e1

Observation state properties

φ: A observes c1 → A1 observes c1
φ: A observes c2 → A2 observes c2
φ: A observes c3 → A3 observes c3

Action initiation state properties

φ: A initiates action a1 → A4 initiates action b1

Mental state property to external world state property

φ: m1 → m2

Note that in this case, for simplicity it is assumed that each observation of A is an observation of exactly one of the Ai, and the same for actions.

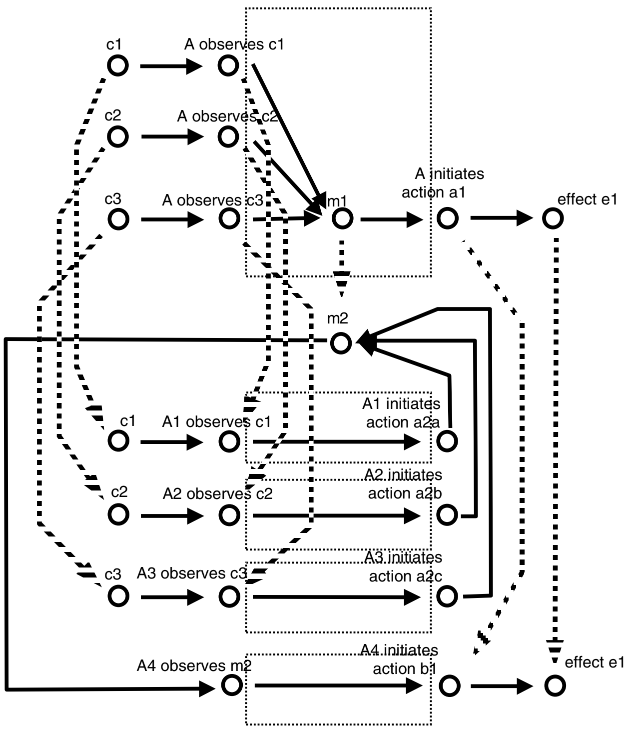


Fig. 3. Isomorphism relationship between shared extended mind and individual mental state

This mapping φ, indicated by the vertical dotted arrows in Figure 3, preserves the temporal dependencies in the form of *leads to* relationships (the solid arrows) and provides an (isomorphic, in the mathematical sense) embedding of a cognitive process based on internal mind into a cognitive process based on extended mind.

In their paper about extended mind, Clark and Chalmers [3] point at the similarity between cognitive processes in the head and some processes involving the external world. This similarity can be used as an indication that these processes can be considered extended cognitive processes or extended mind:

> If, as we confront some task, a part of the world functions as a process which, *were it done in the head*, we would have no hesitation in recognizing as part of the cognitive process, then that part of the world *is* (so we claim) part of the cognitive process. Cognitive processes ain't (all) in the head! [3], Section 2. (…)
>
> One can explain my choice of words in Scrabble, for example, as the outcome of an extended cognitive process involving the rearrangement of tiles on my tray. Of course, one could always try to explain my action in terms of internal processes and a long series of "inputs" and "actions", but this explanation would be needlessly complex. If an isomorphic process were going on in the head, we would feel no urge to characterize it in this cumbersome way. (…) In a very real sense, the re-arrangement of tiles on the tray is not part of action; it is part of *thought*. [3], Section 3.

Clark and Chalmers [3] use the isomorphism to a process 'in the head' as one of the criteria to consider external and interaction processes as cognitive, or mind processes. As the shared mental state property m2 is modelled as an external state property, this 'isomorphism principle' is formalised in Figure 3 for a simple example of such an isomorphism. Note that the process from m1 to action a1, modelled as one step in the single agent, internal case, is mapped onto a process from m2 via A4 observes m2 to A4 initiates action b1, in the external case modelled as a two-step process. So the isomorphism is an embedding in one direction, not a bidirectional isomorphism, simply because on the multi-agent side, the observation state for A4 observing m2 has no counterpart in the single agent, internal case (and the same for the agents A1, A2, A3 initiating actions a2a, a2b, a2c).

Notice that the mapping φ is a (formal) mapping between state properties. However, it was already put forward that temporal *leads to* relations are preserved under φ, so the mapping can be extended to a mapping of *leads to* properties onto *leads to* properties. From a more general perspective, it can be analysed how far the mapping φ can be extended to a (formal) mapping from dynamic properties to dynamic properties expressed in TTL. This will be addressed in detail in Section 6.

4 Two Conceptualisations and Their Mapping

The general formalisation perspective put forward in previous sections has been evaluated for a case study: a process of collective ant behaviour. For this example process two conceptualisations have been made, one from a multi-agent (or social) perspective, and one for a single agent (or cognitive) perspective.

The world in which the ants live is described by a labeled graph as depicted in Figure 4. Locations are indicated by A, B,…, and edges by E1, E2,… To represent such a graph the predicate connected_to_via(l0,l1,e1) is used. The ants move from location to location via edges; while passing an edge, pheromones are dropped. The same or other ants sense these pheromones and follow the route in the direction of the strongest concentration. Pheromones evaporate over time; therefore such routes can vary over time. The goal of the ants is to find food and bring this back to their nest. In this example there is only one nest (location A) and one food source (location F).

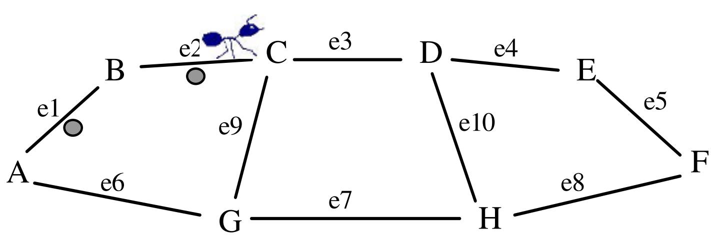


Fig. 4. An ants world

4.1 Multi-agent Conceptualisation

The example process conceptualised from a multi-agent perspective concerns multiple agents (the ants), each of which has input (to observe) and output (for moving and dropping pheromones) states, and a physical body which is at certain positions over

Table 1. Multi-Agent conceptualisation: state properties

	Multi-Agent Conceptualisation
	body positions in world:
pheromone level at edge e is i	pheromones_at(e, i)
ant a is at location l coming from e	is_at_location_from(a, l, e)
ant a is at edge e to l2 coming from location l1	is_at_edge_from_to(a, e, l1, l2)
ant a is carrying food	is_carrying_food(a)
	world state properties:
edge e connects location l1 and l2	connected_to_via(l1, l2, e)
location l has i neighbours	neighbours(l, i)
edge e is most attractive for ant a coming from location l	attractive_direction_at(a, l, e)
	input state properties:
ant a observes that it is at location l coming from edge e	observes(a, is_at_location_from(l, e))
ant a observes that it is at edge e to l2 coming from location l1	observes(a, is_at_edge_from_to(e, l1, l2))
ant a observes that edge e has pheromone level i	observes(a, pheromones_at(e, i))
	output state properties:
ant a initiates action to go to edge e to l2 coming from location l1	to_be_performed(a, go_to_edge_from_to(e, l1, l2))
ant a initiates action to go to location l coming from edge e	to_be_performed(a, go_to_location_from(l, e))
ant a initiates action to drop pheromones at edge e coming from location l	to_be_performed(a, drop_pheromones_at_edge_from(e, l))
ant a initiates action to pick up food	to_be_performed(a, pick_up_food)
ant a initiates action to drop food	to_be_performed(a, drop_food)

time, but no internal mental state properties (they are assumed to act purely by stimulus-response behaviour). An overview of the formalisation of the state properties of this multi-agent conceptualisation is shown in Table 1.

4.2 Single-Agent Conceptualisation

The conceptualisation of the example process from a single agent perspective (Superant S), however, takes into account one body, of which each ant is part (for

Table 2. Single Agent conceptualisation: state properties

Single Agent Conceptualisation	
mental state properties:	
belief(S, relevance_level(e, i))	belief on the relevance level i of an edge e
body position in world:	
has_paw_at_location_from(S, p, l, e)	position of paw p at location l coming from edge e
has_paw_at_edge_from_to(S, p, e, l1, l2)	position of paw p at edge e to l2 coming from location l1
is_carrying_food_with_paw(S, p)	paw p is carrying food
world state properties:	
connected_to_via(l1, l2, e)	edge e connects location l1 and l2
neighbours(l, i)	location l has i neighbours
attractive_direction_at(p, l, e)	edge e is most attractive for paw p coming from location l
input state properties:	
observes(S, has_paw_at_location_from(p, l, e))	S observes that paw p is at location l coming from edge e
observes(S, has_paw_at_edge_from_to(p, e, l1, l2))	S observes that paw p is at edge e to l2 coming from location l1
output state properties:	
to_be_performed(S, move_paw_to_edge_from_to(p, e, l1, l2))	S initiates action to move paw p from location l1 to edge e to l2
to_be_performed(S, move_paw_to_location_from (p, l, e))	S initiates action to move paw p from edge e to location l
to_be_performed(S, pick_up_food_with_paw(p))	S initiates action to pick up food with paw p
to_be_performed(S, drop_food_with_paw(p))	S initiates action to drop food with paw p

convenience we call them the 'paws' of this body). Also the pheromone levels at the edges are part of the body.

The body position of this agent in the world is defined by the collection of positions of each of the paws. Mental state properties for this single agent occur in the form of beliefs that a certain edge has a certain relevance level (realised in the body by the pheromone levels). Input of the single agent is defined by the collection of inputs of the ants at each of the paws. Output is defined by initiation of movements of one or more of the paws. Notice that in this case dropping pheromones is not an action, but an internal body process to create or update the proper beliefs by creating or updating their realisation in the body. An overview of the formalisation of the state properties of the multi-agent conceptualisation is shown in Table 2. Note that there S stands for the Superant.

4.3 Mapping Between Conceptualisations

The two conceptualisations described in Sections 4.1 and 4.2 are two conceptualisations of one and the same example process. A concept in any of the two conceptualisations in principle has a one-to-one correspondence to an aspect of this example process which can be considered the informal semantics of the concept (in our case the concept is formalised); see the double arrows in Figure 5.

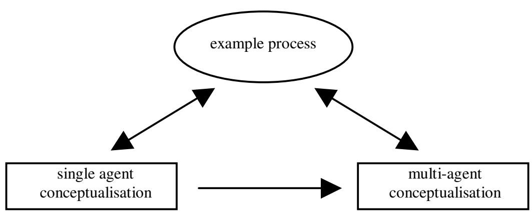


Fig. 5. Two conceptualisations and their mapping

Given these one-to-one correspondences, a mapping from the single agent conceptualisation to the multi-agent conceptualisation can be made as follows:

1) Take any state property c belonging to the single agent conceptualisation
2) Identify to what aspect a of the example process this state property corresponds
3) Identify to which state property d in the multi-agent conceptualisation this aspect a corresponds
4) Map c to d.

If this approach works, then a mapping is obtained that is sincere with respect to the example process: the state property d to which c is mapped corresponds to the same. aspect a of the process as c, and therefore will be true (for the informal semantics) if and only if c is. The approach can also fail. It can fail in 2) if state properties are used in the single agent conceptualisation that have no counterpart in the example process. It can fail in 3) if in the single agent conceptualisation aspects of the process are covered that are left out of consideration in the other conceptualisation. Actually such aspects exist the other way around: there are aspects of the process, such as observing the pheromones covered by the multi-agent conceptualisation, but not by the single

Table 3. Mapping between state properties

Single Agent Conceptualisation	Multi-Agent Conceptualisation
belief(S, relevance_level(e, i))	pheromones_at(e, i)
has_paw_at_location_from(S, p, l, e) has_paw_at_edge_from_to(S, p, e, l1, l2) is_carrying_food_with_paw(S, p)	is_at_location_from(a, l, e) is_at_edge_from_to(a, e, l1, l2) is_carrying_food(a)
connected_to_via(l1, l2, e) neighbours(l, I) attractive_direction_at(p, l, e)	connected_to_via(l1, l2, e) neighbours(l, i) attractive_direction_at(a, l, e)
observes(S, has_paw_at_location_from(p, l, e)) observes(S, has_paw_at_edge_from_to(p, e, l1, l2)) ---	observes(a, is_at_location_from(l, e)) observes(a, is_at_edge_from_to(e, l1, l2)) observes(a, pheromones_at(e, i))
to_be_performed(S, move_paw_to_edge_from_to(p, e, l1, l2)) to_be_performed(S, move_paw_to_location_from (p, l, e)) --- to_be_performed(S, pick_up_food_with_paw(p)) to_be_performed(S, drop_food_with_paw(p))	to_be_performed(a, go_to_edge_from_to(e, l1, l2)) to_be_performed(a, go_to_location_from(l, e)) to_be_performed(a, drop_pheromones_at_edge_from(e, l)) to_be_performed(a, pick_up_food) to_be_performed(a, drop_food)

agent conceptualisation. Therefore such a mapping is not possible from right to left in Figure 5 (see also Figure 3 in Section 3, where the mapping is not bijective either). However, a mapping from left to right (single agent to multi-agent conceptualisation), is possible. It is shown in Table 3. Note that there S stands for the Superant, and paw p corresponds to ant a

5 Two Simulation Models

The two conceptualisations introduced above have been used to create two simulation models for collective ant behaviour: one from a multi-agent (social) perspective and one from a single agent (cognitive) perspective. The basic building blocks of the model were dynamic properties in *leads to* format, specifying the local mechanisms of the process. Examples of such local dynamic properties (for the *multi-agent case*) are the following:

LP5 (Selection of Edge)
"If an ant observes that it is at location l, and there are three edges connected to that location, then the ant goes to the edge with the highest amount of pheromones." Formalisation:
observes(a, is_at_location_from(l, e0)) and neighbours(l, 3) and connected_to_via(l, l1, e1) and observes(a, pheromones_at(e1, i1)) and connected_to_via(l, l2, e2) and observes(a, pheromones_at(e2, i2)) and e0 $\neq$ e1 and e0 $\neq$ e2 and e1 $\neq$ e2 and i1 > i2 $\bullet\!\!\rightarrow$ to_be_performed(a, go_to_edge_from_to(e1, l1))

LP6 (Arrival at Edge)
"If an ant goes to edge e from location l to location l1, then later the ant will be at this edge e."
to_be_performed(a, go_to_edge_from_to(e, l, l1)) $\bullet\!\!\rightarrow$ is_at_edge_from_to(a, e, l, l1)

LP9 (Dropping of Pheromones)
"If an ant observes that it is at an edge e from a location l to a location l1, then it will drop pheromones at this edge e."
observes(a, is_at_edge_from_to(e, l, l1)) $\bullet\!\!\rightarrow$ to_be_performed(a, drop_pheromones_at_edge_from(e, l))

LP12 (Observation of Pheromones)
"If an ant is at a certain location l, then it will observe the number of pheromones present at all edges that are connected to location l."
is_at_location_from(a, l, e0) and connected_to_via(l, l1, e1) and pheromones_at(e1, i) •→ observes(a, pheromones_at(e1, i))

LP13 (Increment of Pheromones)
"If an ant drops pheromones at edge e, and no other ants drop pheromones at this edge, then the new number of pheromones at e becomes i*decay+incr." Here, i is the old number of pheromones, decay is the decay factor, and incr is the amount of pheromones dropped.
to_be_performed(a1, drop_pheromones_at_edge_from(e, l1)) and ∀l2 not to_be_performed(a2, drop_pheromones_at_edge_from(e, l2)) and ∀l3 not to_be_performed(a3, drop_pheromones_ at_ edge_ from(e, l3)) and a1 ≠ a2 and a1 ≠ a3 and a2 ≠ a3 and pheromones_at(e, i) •→ pheromones_at(e, i*decay+incr)

LP14 (Collecting of Food)
"If an ant observes that it is at location F (the food source), then it will pick up some food."
observes(a, is_at_location_from(F, e)) •→ to_be_performed(a, pick_up_food)

To model the example from a single agent perspective, again a number of local dynamic properties are used. Most, but not all of these local properties have a 1:1 correspondence to those for the multi-agent case. For example, the properties for the *single agent case* that correspond to the properties above are as follows (see the next section for more information about this correspondence):

LP5' (Selection of Edge)
"If S observes that it has a paw p at location A, and there are three edges connected to that location, then S will move its paw to the edge of which it believes that it has the highest relevance level."
observes(S, has_paw_at_location_from(p, l, e0)) and neighbours(l, 3) and connected_to_via(l, l1, e1) and belief(S, relevance_level(e1, i1)) and connected_to_via(l, l2, e2) and belief(S, relevance_level(e2, i2)) and e0 ≠ e1 and e0 ≠ e2 and e1 ≠ e2 and i1 > i2 •→ to_be_performed(S, move_ paw_ to_ edge_from_to(p, e1, l1))

LP6' (Paw Arrival at Edge)
"If S moves its paw p to an edge e from a location l to a location l1, then later this paw will be at this edge e."
to_be_performed(S, move_paw_to_edge_from_to(p, e, l, l1)) •→ has_paw_at_edge_from_to(S, p, e, l, l1)

LP11' (Increment of Belief)
"If S has exactly one paw at edge e, then the new number of pheromones at e becomes i*decay+incr."
observes(S, has_paw_at_edge_from_to(p1, e, l, l1)) and ∀l2 not observes(S, has_paw_at_edge_from_to(p2, e, l, l2)) and ∀l3 not observes(S, has_paw_at_edge_from_to(p3, e, l, l3)) and p1 ≠ p2 and p1 ≠ p3 and p2 ≠ p3 and belief(S, relevance_level(e, i)) •→ belief(S, relevance_level(e, i*decay+incr))

LP12' (Collecting of Food)
"If S observes that it has a paw p at location F (the food source), then it will pick up some food with that paw."
observes(S, has_paw_at_location_from(p, F, e)) •→ to_be_performed(S, pick_up_food_with_paw(p))

A special software environment has been created to enable the simulation of executable models. Based on an input consisting of dynamic properties in *leads to* format, it can generate simulation traces. An example of (part of) such a trace can be seen in Figure 6. Time is on the horizontal axis, the state properties are on the vertical axis. A dark box on top of the line indicates that the property is true during that time

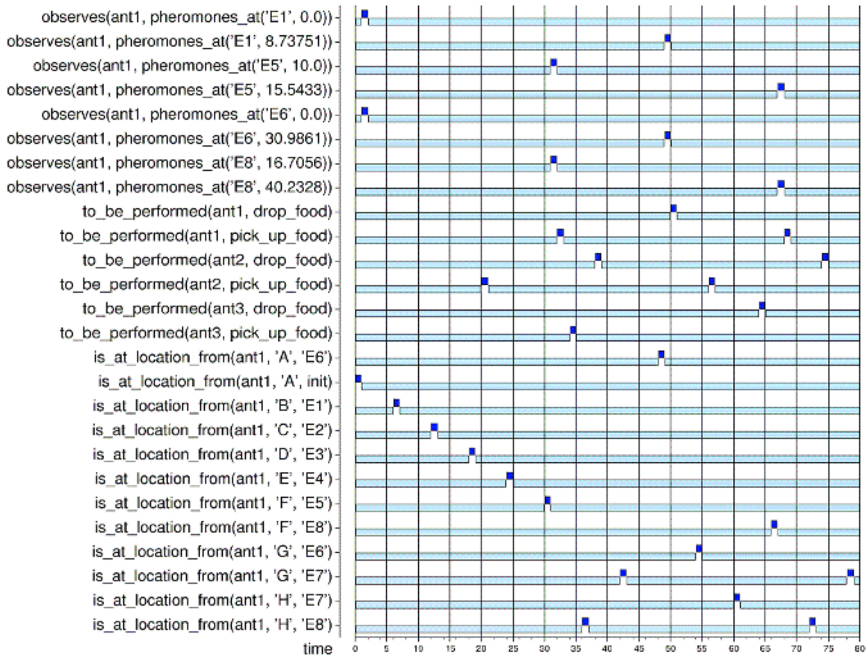


Fig. 6. Multi-Agent Simulation Trace

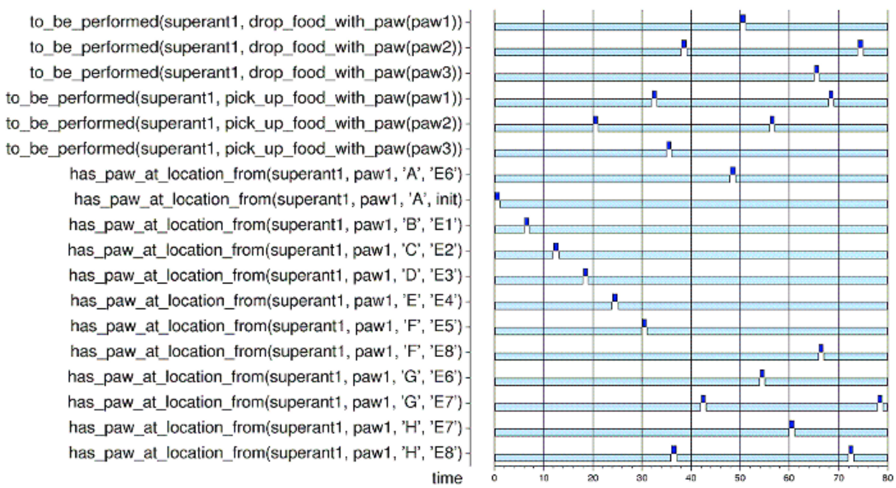


Fig. 7. Single Agent Simulation Trace

period, and a lighter box below the line indicates that the property is false. This trace was based on the multi-agent simulation model.

Figure 7 depicts a similar trace as Figure 6, this time based on the single agent simulation model. Note that there are several differences between Figure 6 and 7. In the first place, all ants that are treated as separate agents in Figure 6, are considered as parts of Superant S in Figure 7. For example, is_at_location_from(ant1, A, E6)) in the multi-agent case corresponds to has_paw_at_location_from(S, paw1, A, E6)) in the single agent case. Another important difference is that in the single agent case, there is no explicit observation of pheromones. The reason for this is that the belief(S, relevance_level(e, i)) states (which are the single agent equivalent for the pheromones_at(e, i) states in the multi-agent case) are internal states of S, which do not have to be observed.

Altogether, the software environment has been used to successfully generate a large number of simulation traces on the basis of both simulation models. To limit complexity, only some fragments of such traces are shown here.

6 The Extended Interpretation Mapping

In Section 3 it was shown how the basic interpretation mapping can be defined as a mapping between state properties. It was suggested that this mapping can be extended to a mapping between local dynamic properties in *leads to* format. Therefore, the following interpretation mapping can be defined:

$$\varphi(\alpha \bullet\!\!\rightarrow \beta) = \varphi(\alpha) \bullet\!\!\rightarrow \varphi(\beta)$$

Using this interpretation mapping, combined with the basic mapping of the state ontology elements described in Section 4, mappings between the dynamic properties of the case study can be found, e.g.:

φ(LP6')
= φ(to_be_performed(S, move_paw_to_edge_from_to(p, e, l, l1)) •→ has_paw_at_edge_from_to(S, p, e, l, l1))
= φ(to_be_performed(S, move_paw_to_edge_from_to(p, e, l, l1))) •→
φ(has_paw_at_edge_from_to(S, p, e, l, l1))
= to_be_performed(a, go_to_edge_from_to(e, l, l1)) •→ is_ at_edge_from_to(a, e, l, l1))
= LP6

A mapping between some of the local dynamic properties (in *leads to* format) of the case study is given in Table 4. Notice that in some cases a certain dynamic property is mapped to a dynamic property that is not literally in the multi-agent model, but actually is a combination of two other local properties present in the model. This shows where the single agent conceptualisation is simpler than the multi-agent conceptualisation.

Table 4. Mapping between local dynamic properties

Single Agent Conceptualisation	Multi-Agent Conceptualisation
LP5'	LP5 & LP12
LP6'	LP6
LP11'	LP9 & LP13
LP12'	LP14

The mapping shown in Table 4 is a syntactic mapping. However, also the traces generated on the basis of these properties can be mapped: each trace γ can be mapped onto a trace $\varphi(\gamma) = \gamma'$. For example, the trace depicted in Figure 7 can be mapped onto the trace depicted in Figure 6. This shows that the syntactic mapping between local properties preserves semantics.

In addition, it is possible to extend the mapping to the wider class of TTL expressions. Recall that TTL expressions are built on atoms of the form state(γ, t) |= p. By the basic mapping the state property p can be translated into φ(p), which is assumed to be part of the ontology of one of the agents Ai in the multi-agent conceptualisation. Moreover, the trace name γ can be mapped onto a trace name φ(γ) = γ'. Then the extended interpretation mapping for state(γ, t) |= p is defined by:

$$\varphi\text{: state}(\gamma, t) \models p \;=\; \text{state}(\gamma', t) \models \varphi(p)$$

After these atoms have been mapped, TTL expressions as a whole can be mapped in a straightforward compositional manner:

$$\varphi(A\&B) \;=\; \varphi(A)\ \&\ \varphi(B)$$
$$\varphi(A \Rightarrow B) \;=\; \varphi(A) \Rightarrow \varphi(B)$$
$$\varphi(\text{not } A) \;=\; \text{not } \varphi(A)$$
$$\varphi(\forall v\ A(v)) \;=\; \forall v'\ \varphi(A(v'))$$
$$\varphi(\exists v\ A(v)) \;=\; \exists v'\ \varphi(A(v'))$$

For example, take the following TTL expression, which is a global property for the single agent case of the ant example:

GP1' Food Discovery

"Eventually, one of the paws of S will be at the food location."

∃t,p,l,e [state(γ, t) |= has_paw_at_location_from(S, p, l, e) & state(γ, t) |= food_location(l)]

This expression is mapped as follows:

φ(∃t,p,l,e [state(γ, t) |= has_paw_at_location_from(S, p, l, e) & state(γ, t) |= food_location(l)])

= ∃t',p',l',e' φ([state(γ, t') |= has_paw_at_location_from(S, p', l', e') & state(γ, t') |= food_location(l')])

= ∃t',p',l',e' [φ(state(γ, t') |= has_paw_at_location_from(S, p', l', e')) & φ(state(γ, t') |= food_location(l'))]

= ∃t',p',l',e' [state(γ', t') |= φ(has_paw_at_location_from(S, p', l', e')) & state(γ', t') |= φ(food_location(l'))]

= ∃t',p',l',e' [state(γ', t') |= is_at_location_from(p', l', e') & state(γ', t') |= food_location(l')]

Thus, eventually global property GP1' is mapped to the folling global property (GP1):

GP1 Food Discovery

"Eventually, one of the ants will be at the food location."

∃t,a,l,e [state(γ, t) |= is_at_location_from(a, l, e) & state(γ, t) |= food_location(l)]

7 Discussion

This paper addresses the question to what extent a process involving multiple agents that shows some form of collective intelligence can be interpreted as single agent behaviour. The question is answered by formal analysis. It is shown for an example process how it can be conceptualised and formalised in two different manners: from a single agent (or cognitive) and from a multi-agent (or social) perspective. Moreover, it is shown how a basic ontological mapping can be formally defined between the two formalisations, and how this mapping can be extended to a mapping of dynamic properties. Thus it is shown how the collective behaviour can be interpreted in a formal manner as single agent behaviour. For example, the fact that food is taken from the source to the nest can be explained by a sequence of actions of one agent, based on its beliefs.

Having such a mapping allows one to explain collective or social behaviour in terms of single agent concepts in the following manner. Behaviour often is explained by considering the basic underlying causal relations or mechanisms. The mapping and its formalisation allows to replace an explanation of behaviour in terms of basic mechanisms involving frequent interactions of the multiple agents (with each other and/or with the external world), by an explanation that leaves out these interactions and bases itself directly on mental states of the single agent conceptualisation. This explanation is often simpler, more abstract, better understandable, and perhaps more elegant than the more complicated explanation based on the interactions. This is made possible by introducing a new ontology for states involved. For example, considering part of the external world as extended mind allows one to give another interpretation to external physical processes and states. Physical state properties such as 'pheromone is present at d' are reconceptualised as, for example, 'it is believed that d is a relevant path'.

Why would one introduce extra language to refer to the same fact in the world? Given the literature on reduction, where often it is claimed that mental state properties can be and actually should be replaced by their physical realisers, at first sight such an opposite move may seem a bit surprising. For example, Kim [12] (pp. 214-216) claims that ontological simplification is one of the reasons to reduce mental state properties to physical state properties. In the extended mind case at hand the converse takes place; a question is what is the advantage of this ontological complication. A number of arguments in support of this can be given. By Clark and Chalmers [3], it is claimed that this allows application of other types of explanation and other methods of scientific investigation:

> (…) we allow a more natural explanation of all sorts of actions. (…) in seeing cognition as extended one is not merely making a terminological decision; it makes a significant difference to the methodology of scientific investigation. In effect, explanatory methods that might once have been thought appropriate only for the analysis of "inner" processes are now being adapted for the study of the outer, and there is promise that our understanding of cognition will become richer for it. [3], Section 3.

In [11] it is explained in some detail why in various cases in other areas (such as Computer Science) such an antireductionist strategy often pays off; some of the discussed advantages in terms of insight, transparency and genericity are: additional higher-level ontologies can improve understanding as they may allow simplification of the picture by abstracting from lower-level details; more insight is gained from a conceptually higher-level perspective; analysis of more complex processes is possible; finally, the same concepts have a wider scope of application, thus obtaining unification.

Also by Dennett it is claimed that the use of a different ontology for the same world facts can be beneficial. In [7], he puts forward the *intentional stance*, a perspective that allows one to describe certain physical phenomena in terms of mental concepts such as desires and intentions, in order to obtain more understandable explanations:

> Predicting that someone will duck if you throw a brick at him is easy from the folk-psychological stance; it is and will always be intractable if you have to trace the protons

from brick to eyeball, the neurotransmitters from optic nerve to motor nerve, and so forth. [7], p. 42.

In this context, the perspective taken in the current paper can be viewed as an extension of the intentional stance, where mental concepts are ascribed not only to single agents, but also to processes that can be conceptualised as groups of agents.

Given this perspective, the question might come up whether the behavioural description of the resulting ‘super-agent’ will not become just as complex as that of the initial multi-agent system. Two answers may be given to this question. First, the case study addressed in this paper has shown that there are at least a number of concepts in the multi-agent description that can be left out in the single agent description. In particular, such concepts are the creation and the observation of the ‘shared extended mental state’ (state m2 in Figure 3); also see Table 3 where for some concepts in the right column there is no counterpart in the left column. Moreover, even if the single agent description is still rather complex, this does not have to be a problem. Within cognitive science, many approaches exist to handle complexity of an agent’s mental processes by imposing structure on it, see, e.g., [9]. In this view, the collective behaviour of a group of agents may be seen as single agent behaviour that consists of a number of sub-processes.

In Section 4.3, it was mentioned that the mapping from multi-agent to single agent conceptualisation is unidirectional, not bidirectional. The main reason for this was that a number of the ‘collective’ concepts did not have an ‘individual’ counterpart. However, in the literature on philosophy of mind, several authors show that in some cases it might also be beneficial to explain an *individual* mental process as a *collective* process (see, e.g., [5]). Thus, it might be useful to explore more possibilities to obtain a mapping in the opposite direction. In future work, these possibilities will be investigated in more detail.

Other future research will further analyse the interpretation mapping in the context of logic: the notion of an interpretation of one (formal) logical theory T in another logical theory T' has a formal definition in logic. It is an interesting question whether it can be proven logically that the conditions of this definition are fulfilled for the mapping defined in this paper. For example, a question is whether it can be proven that:

$$T \vdash \alpha \Rightarrow T' \vdash \varphi(\alpha)$$

for all formulae α, where T is a logical theory of single agent behaviour and T' a theory of multi-agent behaviour. More specifically, suppose that a global property B is implied by a number of local properties $A_1, \ldots, A_n$, according to the following relation:

$$A_1 \ \& \ \ldots \ \& \ A_n \Rightarrow B$$

Given this implication, the question to explore would be whether there is a similar relation available between the mapped properties, i.e., whether the following implication:

$$\varphi(A_1) \ \& \ \ldots \ \& \ \varphi(A_n) \Rightarrow \varphi(B)$$

holds as well.

Finally, the authors would like to stress that the main aim of the current paper was to show *how* a process involving multiple agents can be interpreted as single agent behaviour, assuming that this is possible. In future work, more attention will be paid to the question *when* a process involving multiple agents can be interpreted as single agent behaviour, i.e., which criteria should hold in order for the mapping to be useful.

Acknowledgements

The authors are grateful to Catholijn Jonker for many valuable discussions on the topic of the paper, to Lourens van der Meij for his contribution to the development of the software environment, and to Martijn Schut for his contribution to parts of the simulations.

References

1. Bonabeau, E., Dorigo, M., and Theraulaz, G. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York, 1999.
2. Clark, A. *Being There: Putting Brain, Body and World Together Again*. MIT Press, 1997.
3. Clark, A. and Chalmers, D. The Extended Mind. In: *Analysis,* vol. 58, 1998, pp. 7-19.
4. Deneubourg, J.L., Aron, S., Goss, S., Pasteels, J.M., and Duerinck, G. Random Behavior, Amplification Processes and Number of Participants: How They Contribute to the Foraging Properties of Ants. In: *Evolution, Games and Learning: Models for Adaptation in Machines and Nature*, North Holland, Amsterdam, 1986, pp. 176-186.
5. Dennett, D.C. *Consciousness Explained,* Little, Brown: Boston, Massachusetts, 1991.
6. Dennett, D.C. *Kinds of Mind: Towards an Understanding of Consciousness*, New York: Basic Books, 1996.
7. Dennett, D.C. Real Patterns. *The Journal of Philosophy*, vol. 88, 1991, pp. 27-51.
8. Drogoul, A., Corbara, B., and Fresneau, D. MANTA: New experimental results on the emergence of (artificial) ant societies. In: Gilbert N. and Conte R. (eds.), *Artificial Societies: the computer simulation of social life*, UCL Press, 1995.
9. Fodor, J.A. The *Modularity of Mind,* Bradford Books, MIT Press: Cambridge, Massachusetts, 1983.
10. Jonker, C.M., Treur, J., and Wijngaards, W.C.A. A Temporal Modelling Environment for Internally Grounded Beliefs, Desires and Intentions. *Cognitive Systems Research Journal*, vol. 4(3), 2003, pp. 191-210.
11. Jonker, C.M., Treur, J., and Wijngaards, W.C.A. Reductionist and Antireductionist Perspectives on Dynamics. *Philosophical Psychology Journal*, vol. 15, 2002, pp. 381-409.
12. Kim, J. *Philosophy of Mind*. Westview Press, 1996.
13. Kirsh, D. and Maglio, P. On distinguishing epistemic from pragmatic action. *Cognitive Science*, vol. 18, 1994, pp. 513-49.
14. Menary, R. (ed.). *The Extended Mind,* Papers presented at the Conference *The Extended Mind - The Very Idea: Philosophical Perspectives on Situated and Embodied Cognition*, University of Hertfordshire, 2001. John Benjamins, 2005, to appear.

An Artificial Maieutic Approach for Eliciting Experts' Knowledge in Multi-agent Simulations

François Sempé[1], Minh Nguyen-Duc[1,2], Stanislas Boissau[3], Alain Boucher[1], and Alexis Drogoul[4]

[1] Institut de la Francophonie pour l'Informatique (IFI), Hanoi, Vietnam
francois@ifi.edu.vn
alain.boucher@auf.org
[2] Laboratoire d'Informatique de l'université de Paris 6 (LIP6), Paris, France
minh.nguyen-duc@lip6.fr
[3] Wageningen Univerty, Netherlands, and Ecole des Hautes Etudes en Sciences Sociales, Paris, France
sboissau@fpt.vn
[4] Institut de Recherche pour le Développement (IRD), Bondy, France
alexis.drogoul@bondy.ird.fr

Abstract. Models of human behaviours used in multi-agent simulations are limited by the ability of introspection of the social actors: some of their knowledge (reflexes, habits, non-formalized expertise) cannot be extracted through interviews. The use of computer-mediated role playing games put these actors into a situated stance where the recording of their "live" behaviours is possible. But cognitive processes and motivations still have to be interpreted.

In this paper, we propose an artificial maieutic approach to extract such pieces of knowledge, by helping the actors to better understand, and sometimes formulate, their own behaviours. The actors are playing their own roles in an agent-mediated simulation and interact with agents that question their behaviours. The actor's reactions and understanding are stimulated by these interactions, and this situation allows in many cases to reveal hidden knowledge. We present here the first results using two complementary works in social simulations, one in the domain of air traffic control and one in the domain of common-pool resources sharing.

1 Introduction

Predicting the evolution of a social organisation in which a number of people are involved in collective (collaborative or concurrent) activities requires to undertake simulations. One of their most common applications is to evaluate the outcomes of new management policies or working procedures before they are applied, especially in critical domains where security is essential, like air traffic control. In this respect, agent-based techniques are broadly used to simulate human beings thanks to the inherent "social" nature of agents, which enables modellers to easily represent in an artificial environment processes like interaction, communication, or collaboration between people.

J.S. Sichman and L. Antunes (Eds.): MABS 2005, LNAI 3891, pp. 75 – 87, 2006.

In order to create agents that represent human beings, however, it is necessary to model relevant parts of human knowledge into sets of behaviours, decision rules, or heuristics which will be available for each agent. When the knowledge is already available through previous (sociological, anthropological, ethnological) studies, the translation into a computational model, although it can take some time, is mostly a matter of finding an appropriate supporting architecture. When it is not available, it requires to be extracted from experts or directly from actors of the target human organisation, through interviews, inquiries, or field experiments. However, most extraction methods encounter a number of difficulties when it comes to model informal knowledge that come from the experience of the actors rather than through classical learning.

We propose to tackle this issue by using simulation as a support for building such models, thanks to a computer mediated, continuous dialogue between human actors and artificial agents [6]. Our hypothesis is that the actor can more easily describe his behaviour or knowledge when placed in situation in a role-playing game supported by a simulation. In what we call an artificial maieutic approach, the agent questions the actor and tests his reactions, either directly ("Why such an action?") or indirectly (through a modification of the perceptions available) in order to explore his informal knowledge.

Two experiments are presented in this paper in order to illustrate our approach, the first one in the domain of air traffic control and the second one in social sciences.

2 Traditional Social Simulation

The concept of "multi-agent system", since its emergence in the 1980s, has always been considered, among other things, as an interesting modelling and simulation tool for social sciences. In the 1990s, if we set aside "toy simulations" serving as illustrations of social theories (like Sugarscape for instance), social multi-agent simulation began to be used in critical domains like military [8] or industrial research (a good example is the simulation of a large population of consumers made by France Telecom in [13]).

More recently, and in a similar way to what we propose in this paper, researchers from social and agronomic sciences working on the management of renewable resources have explored agent-based approaches to model and understand the outcome of, for instance, different sharing policies on the availability of resources [2]. Their goal is to use multi-agent simulations, simultaneously as a support for experimental research and as a computer-aided training and decision-making tool for actors.

3 Artificial Maieutics for Eliciting Informal Knowledge

In order to make such simulation, it is necessary to model human behaviour. Usually, the model is built through an iterative process with the help of interviews between a modelling expert (the one who builds the model) and some social actors. This method is limited by the ability of actors to describe and explain their actions. But a part of the knowledge to be elicited from actors is not accessible through interviews or more general inquiries (sociological or anthropological). Reflexes, habits, reactions to unexpected situations or behaviours refined by experience represent some informal knowledge hard to capture and to formalize. How can we have access to this kind of knowledge in order to improve the models?

In this paper, we got inspiration from *human-in-the-loop* experiments which directly integrate human actors in the running loop of a simulation with the help of a dedicated interface. Our approach relies on the following assumption: immersed in a (simulated) real-conditions situation where he is asked to play his own role, the actor can more naturally exhibit these informal behaviours, and hopefully better understand and describe them to the modellers. In turn, using the simulation as a support enables the modellers to be more accurate when trying to refine the behaviours programmed: as a matter of fact, the questions they ask to the actor are grounded in real situations and they can ask them as soon as this situation appears within the simulation.

We call this approach "artificial maieutics" in relation to the questioning method used by Socrates (Greek philosopher of the 5th century B.C.) to make his interlocutor discover by himself some non-conscious knowledge. In artificial maieutics, the role of the questioner can be played either by an expert or, more interestingly, by an artificial agent during the simulation. In other words, an agent can be attached to the interface as an assistant, which role is to interrogate the actor about his actions, and to record and use the answers in order to improve the model. These two methods share the same basis (illustrated on Figure 1): The first step is to develop a role-playing game where the social actors are asked to play their role in carefully chosen scenarios; this should result in a set of logs or traces, which, in the second step, will be used (either manually or automatically) to program artificial agents.

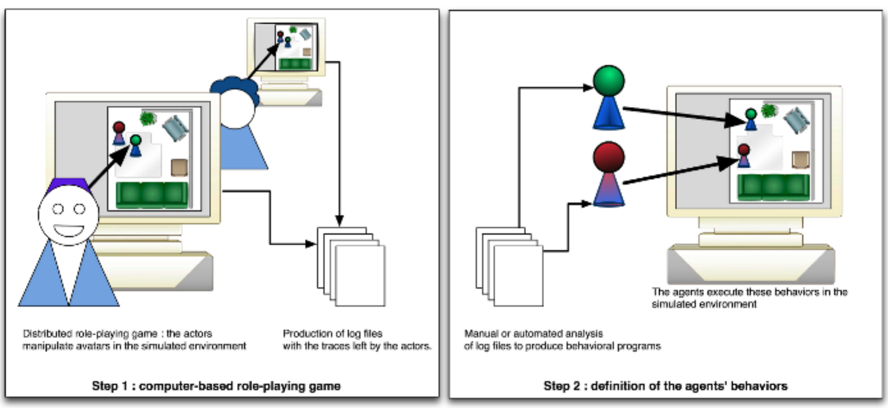


Fig. 1. The first two steps of artificial maieutic experiments

Once the simulation (even with minimal behaviours) is up and running, it can be used as a basis for the maieutic step. Two methods are then available: one which involves active discussions between the actor and an expert, the other active interactions between the actor and "his/her" agent. The first one is usually used when the domain is not formalized or when the goal of the simulation is not sufficiently defined. The second strongly relies on the knowledge available about the domain: the agents have to be aware of the goal of the simulation to understand the answers of the

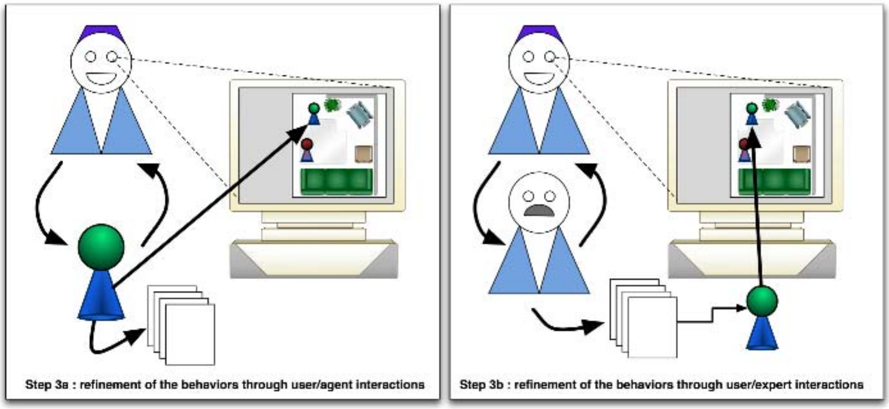


Fig. 2. The two different methods of artificial maieutics

actors. These two methods (see Figure 2) are not antinomian and can be used in sequence (or in parallel) to refine the behaviours of the agents and build a better model.

In order to illustrate the use of these two methods, we present two experiments. The first one shows the use of the user/expert interactions : a group of students participate in a role playing game about common-pool resources sharing and are asked to build and discuss with their teachers a model of their own behaviours.

The second one describes how interface agents can be used for questioning air traffic controllers engaged in a simulation of their daily professional activity (these experiments have been used to validate innovative organizational methods among controllers [11]).

4 Artificial Maieutics and Self-modelling

In this experiment, some computer science students were asked to make a model of their own behaviour during a role playing game. We present here the first step of these experiments where students of the French-speaking institute of computer science (IFI - Institut de la Francophonie pour l'Informatique) in Hanoi, Vietnam, have no artificial maieutic tools to complete the modelling task. This experiment has two purposes. First, it will allow us to better understand the difficulties that someone has to face when making a model of himself: we expect to learn more on how to design these tools. Second, this experiment will be used later as a reference for an evaluation of the artificial maieutic tools.

4.1 Experiment Description

The Settings. The IFI students who took part in this experiment are graduated computer engineers (5 years of university). Some of them are teaching at the university or working in a company. Their abilities in computer science allow them to process the whole modelling task, until the coding of an agent that reproduces their own behaviours.

The modelling workshop was divided into 5 steps that took place during a week:

1) Explanation of the game and play a game.
2) Just after the end of the game, students were asked to describe their behaviour without any help.
3) The day after step 2, the students corrected their first descriptions with the help of a tool to replay their game.
4) From the previous written description, a computational model was built through the writing of pseudo-code and code for an agent that model their behaviour during the game.
5) Finally, the students had to evaluate their model comparing the real game and a simulation of the game running with the agents they coded.

The Game of Friends. The students took part in a role playing game which is similar to another one played by farmers from northern Vietnam (the game of Buffalos) as part of political science research. This research program is focused on people's behaviour when they face the growing scarcity of common-pool resources [3]. In this paper, we are only interested in modelling questions and we will not tackle the social side of the experiment.

The rules of the two games are the same, only the scenario is changed. The game of Buffalos does not suit student players because they are not familiar with the life of a farmer. Unfamiliar situation could lead to fanciful behaviours.

At the beginning of a game, one player starts with a certain amount of friends. At every loop, he has to download a movie on internet (which is against laws of copyright and we do not recommend) for each friend in order to keep him satisfied. During the game, the total number of possible downloaded movies decreases. A friend may not receive his movie, which turns him angry. Angry friends may leave the player. At each loop, the player can make new friend (spending "free time" currency) or leave some old friends. Download resources being shared by all players and the number of friends of one player influence the state of the whole system.

The game has been implemented as client/server application. One client is an interface for a human player or an agent that acts like a human player. Thus it is possible to mix agent players and human players in the same game.

One of the originality of this game lies in the lack of goal: players are free to choose their own goal, there is no winner nor looser. At contrary, most economical games have explicit goals like the maximisation of a profit [12]. In other respects, the student taking part in the game of Friends received no information about it before the beginning of the workshop. Thus, they had no means to prepare themselves in any way, for instance, in building a strategy that would be easy to implement into an agent. The lack of goal for the game and information about the workshop serves the purpose to avoid ad hoc strategies.

23 students took part in the experiments. One game is played by 5 players, humans and artificial and lasts 25 loops. 2 games have artificial players, without the knowledge of human ones. All games took place at the same time, in the same room, but players did not know with whom there were playing.

4.2 Results

Game's Dynamics. First, let us notice the existence of 2 steps during a game. From the beginning to about the sixth loop, resources are in excess and all friends can be satisfied. From the sixth turn, resources were getting scarce, and the game entered in a step of shortage. Then, depending on players actions, the shortage stay high (Figure 2) or the number of resources and the number of friends reach an approximate equilibrium.

The game of friends is thus characterized by crises (shortage) and the need for the players to adapt their behaviours.

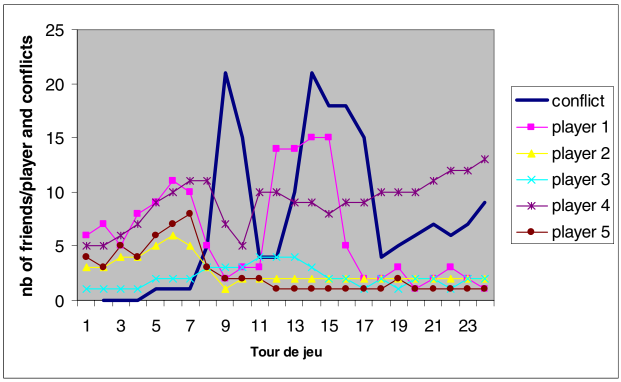


Fig. 3. This chart shows the evolution of the number of friends for each player during 2 games. In this game, the « selfish » behaviour of one player – keeping a lot of friends - maintains the number of conflicts (i.e. the number of unsatisfied friends) at a high level.

Modelling Effect of Coding. In written descriptions of the students' behaviours, we found two opposite faults: the lack of precision ("when conflicts for resources are few…") and the lack of synthesis ("on loop n° 4, I made a new friend…"). Forgetting occurs too: one explains when he makes new friends but tells nothing on the conditions of quitting a friend. In other respects, a lot of description refers to random. "I choose internet sites at random", "I quit friends at random"… Well, in most of the cases, they did not act at random at all as they say. It is enough to watch the game log to be convinced for that. The extensive use of the expression "at random" has the purpose to cover the ignorance of a player who cannot explain why he acts in a certain way. As students had no time to prepare a strategy before the game starts, non-conscious behaviours - hard to describe - have emerged.

The written description was followed by the writing of the pseudo-code for the agent that reproduced their behaviour in the simulation. As pseudo-code contains the logic of a program (tests and loops), this step stands for a hidden modelling process. It has had a clarifying effect: students were forced to transform somehow their imprecise written description into a computational model of their behaviour. For instance, the condition "when resources are getting scarce… » become « when there less than 2 unused resources. This transformation is crucial to the self-modelling process. Let us focus on it.

Creativity in Modelling. It appears that the modelling process use both recollection and creativity. As we explained before, the scenario of the game of Friends has 2 steps. For most of the players this configuration has induced a behaviour switch: the number of friends is increased during the first step when resources are abundant, and reduced during the scarce period. What conditions were chosen to trigger the switch?

In one third of the cases, students chose a condition based on the state of their friends, for instance "if one of my friend was unsatisfied during the last turn, then I do not make new friends". It is a plausible condition as the player could consciously had this perception during the game. But in most of the cases, students had to use their imagination. The one who writes "If more than 80% of internet sites are overused…" did not count the sites during the game, but create a condition that could match his behaviour. One of the students describes explicitly how the recollection of a feeling is turned into model: "I remember getting bothered when resources became rare and I had to change my strategy. A statistical analysis of logged data allows me to discover when I have modified my behaviour, i.e. when the number of resources is less than the total number of friends."

Tendency to Idealization. As they discover some weaknesses or contradictions in their strategy during the game, a lot of students have the tendency to "improve" their behaviour in the model. They want a consistent behaviour for their agent, although we were repeating all time long that in the game of Friends there are no good or bad behaviour. This idealization can be noticed in such commentary: "I could not play like I intended to because I had very few friends at the beginning of the game" – which is obviously hindsight thought.

Tendency to idealization of behaviour is very problematic as it is judgement itself that is biased. "Is this or not a good and useful model?" becomes "Is it or not a nice model of myself" The students did not want to look stupid in this image of themselves, as anybody else would.

During this workshop, students both had to deal with easy-to-model behaviour ("I dot not like to have angry friends so I quit them") and hard-to-model ones ("I have been bothered…", "I played at random…"). Some of those non-conscious behaviours could have been rediscovered through game log analysis, but not all of them, and even when possible, students did not necessarily understand the reason of a given action. Analysis is not enough for understanding. In addition, tendency to idealization of behaviour may bias the whole self-modelling process.

5 A Simulation Tool for Air Traffic Controllers

The second experiment takes place in a specialized domain, where the interactions between social actors are most of the time strictly formalized. Our aim is to show that, thanks to this formalization, artificial agents can be advantageously used for extracting knowledge about the behaviours of the actors in situations impossible to create in real-life experiments. Our tool is based on a multi-agent simulator already implemented for ATC by EuroControl, in which interface agents and functionalities for modelling human behaviour were added.

5.1 Overview of Air Traffic Control

Current Air traffic Management (ATM) system is airspace-based. The airspace is divided in several sectors, the size of which depends on the number of aircraft in the region and the geometry of air routes. There are usually two air traffic controllers to handle the traffic in each air sector: a planning controller and an executing controller. The planning controller works at a strategic level to minimize the number of conflicts or their complexities. The executing controller works at a tactical level to ensure that there are no conflicts, *i.e.* infringements of standard separation between aircrafts by giving instructions to the pilots.

ATM still comprises a higher level of management, *i.e.* the traffic flow management. The flow managers are located in the Central Flow Management Unit and in each control centre. Their task is to (re-)plan the flights at the multi-sector level with as major objective to avoid the congestion and the overload of the controllers (due to the big number of aircrafts to be controlled).

5.2 Agent/Human Hybrid Simulation for an Operational Procedure

A major concern in leaving some loose end to ATM rules is the occurrence of uncontrolled traffic peaks at the entry of a congested area. This phenomenon, often caused by some aircraft “in bunch”, is known in the operational world as “traffic bunching” effect. A way to solve the problem is to structure and organise the arrival flows in real-time. A possible technique is the readjustment of the arrival time of some aircraft at a congested point, thus enabling to “de-bunch” problematical delivery. This technique should enable several controllers and flow managers to collaborate on the traffic for “smoothing” the bunching peaks before they affect the congested area. In our vision, this working group is a kind of social organisation that we can model and simulate.

The investigations undertaken by EuroControl (a European research center in the field of air control) on this collaborative operational procedure need simulation tools able to validate new team-working methods and to be used as support of demonstration bound for real air traffic controllers and traffic flow managers. For this goal, we are implementing an agent/human hybrid simulator in which the human actors (controllers or flow managers) and their assistant agents, are working together like team-mates. We model the interaction between the artificial agents by using STEAM (Shell for TEAMwork), a generic teamwork model described in Tambe et al. [10] The participation of domain experts (*i.e.* controller or manager of flow) is regarded as new for the multi-agent simulations [7] but human-in-the-loop experiments are already largely used at EuroControl for a long time. This participation makes it possible to experiment the maieutic approach of human behaviour modelling.

5.3 Agent/Expert Dialogue

We added to the user interface dedicated to each participant expert an interface agent playing the role of assistant. This last one can play alone the role of the corresponding expert or only assist him. An expert and his assistant constitute, with respect to the other players, one and only one expert/assistant player. The expert plays the role, which was assigned to him within the simulation. The assistant observes, then proposes some behaviours which can be amended by the expert, these modifications being taken into account by the assistant as well as the results of his observations. This agent/expert relation leads to a dialogue in which the assistant questions "why don't you do that for this reason?" and the expert answers by modifying the behaviour suggested "I modify your proposal because of these entities and for this reason". The answer helps to improve the existing human behaviour model. This improvement can be made either by hand by the designer after having studied the log of the simulation game or automatically by agents. However the suitable learning techniques remain to be discovered so in this paper we tackle only the first possibility.

5.4 An Air Traffic Control Simulation Game

To go more into the details of some maieutic dialogues already carried out, we describe here a particular simulation game. We define, in agreement with the experts, a coordination protocol between flow managers in order to create the simulation game. We suppose an initial situation like the following one: a flow manager called "requestor" detects a risk of "traffic bunching" about an hour before it affect a congested sector; this risk is caused by aircraft flying "in bunch" which will cross successively the airspace zones managed by other managers called "suppliers"; the "requestor" informs these "suppliers" of the risk and starts a session of common tactic establishment. The two roles defined here are not exclusive, *i.e.* a flow manager can be at the same time "requestor" and "supplier".

The defined coordination protocol is as the following:

1) The "requestor" builds a pre-tactic to solve this "traffic bunching" risk. This pre-tactic is divided into several measures, each of which is dedicated to handling of one of the aircraft flying "in bunch" and managed by a "supplier". Then it diffuses this pre-tactic to all the "suppliers".
2) Each "supplier" accepts, refuses or modifies associated measures, then diffuses his ideas to the "requestor" and to all the other "suppliers".
3) After having received all the ideas, the "requestor" sees whether there is a refusal. If yes, this coordination failed; if no, it updates and then validates the final common tactics.

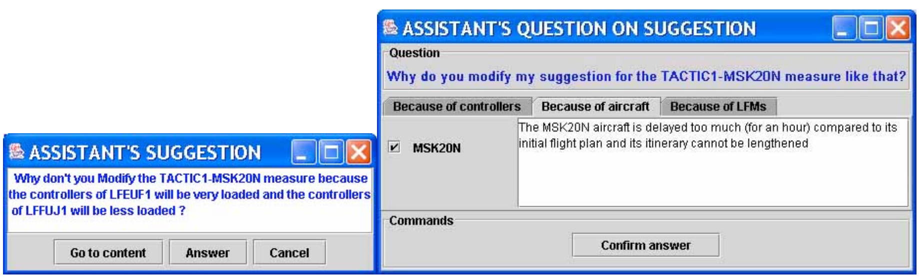
The behaviours of the actors presented above are modelled by the different agents plunged in a similar situation (in the context of pre-established scenarii). The goal is to provide to the experts an intuitive vision of these coordination protocols, and to ensure that they can act easily to modify them, in particular on the following points:

1) **action:** all actions the managers can do in the protocol, *e.g.*, build a pre-tactic, diffuse the pre-tactic, accept a measure, refuse a measure, modify a measure, diffuse the ideas and validate the final tactics.

2) **perception:** all information and data which the managers have to perceive in order to make decisions, *e.g.*, the number of aircrafts present in each sector, geographical information about the sectors and beacons.
3) **reason to act:** the reason for which a manager chooses a specific action and not others, or the reason for which it refuses a measure.

An interaction session between an assistant and an expert player can be described as follows (Figure 4):

1) At one time, the expert has to decide to accept, to refuse or modify a measure, *e.g.*, the re-routing of aircraft MSK20N with the new route segment from beacon RBT to beacon LFMN.
2) The assistant suggests the **action** "modify the measure" which changes the new route segment, *e.g.* it adds beacon ALBET between two beacons AMFOU and ARMUS by explaining that the controllers of sector LFEUF1 will be "very loaded" while those of sector LFFUJ1 will be less "loaded" (Figure 4a).
3) The expert amends this **action** by indicating another new route segment, *e.g.* it takes again the direct way path from AMFOU to ARMUS and removes beacon SINRA.
4) The agent asks for **reasons to act** by presenting a question "Why do you modify my suggestion?"
5) The expert answers this question by describing his **perception**. He chooses the controllers, aircraft or flow managers who cause the amended action and by specifying a reason for each selected controller, aircraft or flow manager. For example, the expert chooses the aircraft MSK20N as the principal cause, and the specified reason is that this aircraft is already "too" delayed (more than one hour) compared to its initial flight plan and that its route cannot be more lengthened (Figure 4b).



(a) (b)

Fig. 4. Examples taken from a session of interaction between an assistant and an ATC actor. (a) Suggestion of the assistant with justification. (b) Question of the assistant on the amendment and answer of the expert.

5.5 Example of Extracted Behaviour

In interaction with the expert, the assistant seeks to build and structure a log of "Actions" being based on the **actions** validated or amended by the expert. This log is used as a data source for the design of the simulator, both in a manual mode, because its structure is designed to help the designer formalizing the behaviours of the agents, and in an automatic mode, with objective to allow the agents to learn by themselves the pertinent behaviours. We present below the structure of an "Action" in the log:

By applying this model at the time of the interaction session described above, the assistant will thus have conserved an "Action" with the following parameters:

Action ≡ (*Predicted_Situation_ assistant, Suggested_Action_Type*, *Suggested_Action*, *Causes_suggestion*, *Reason_suggestion, Amended_Action_Type*, *Amended_Action*, *Causes_amendment*, *Reason_amendment*)

The designer would thus be based on the initial flight plans, on the geography and on the "Actions" conserved in the log to formalize agent behaviours. For example, he can add the following abstract rules:

1) If the difference between the maximum control capacity of a pair of controllers and the number of aircrafts present in their sector is lower than 3, these controllers will be considered as "very loaded".
2) If an aircraft is delayed for more than one hour compared to its initial flight plan, it will be considered as "too delayed".
3) The route of a "too delayed" aircraft cannot be more lengthened even if the lengthening of the route allows it to avoid the "very loaded" controllers.

And new "Rules of deduction" are created from these abstract rules.

6 Discussion and Conclusion

In the domain of political and social sciences, role games and multi-agent simulation have various goals, for example the study or validation of models [2], or the support for negotiation between actors [5]. The usefulness of assistant agents has already been stress, for instance in [4] where the authors want to identify the regularity of interactions between actors.

In this paper, role playing games and multi-agent simulation have been used to propose a new scheme for knowledge acquisition from a human actor to an artificial agent. In other words, this represent a different way of the old questions of how can we give knowledge to an agent and how does the agent can reason using this knowledge? We state in these simulations that we cannot give in one shot all the necessary knowledge to an agent, but only a part of it. This remaining knowledge, coming mainly from human experience, and human instinct, cannot be given so easily. Instead of giving the knowledge straight to the agent, we are giving tools to the agents to observe and step-by-step learn by assisting the human. Instead of resulting in an intelligent systems ready-to-work, the system "only" intend to observe and learn, without short term autonomous actions

In this paper, two complementary works about artificial maieutic were presented. First, in the air traffic simulation, the assistant agent has the possibility to question its own model with the decisions taken by the expert. The questions reflect missing parts in the agent's model compare to the expert's model. This missing part has to be filled in some ways, or by internal update from the agent itself or from update from the actor.

The game of Friends experiment is more prospective. We have shown how difficult it is for an actor to understand and explain even simple actions (making new friends or quitting old ones) made few hours ago. In addition, actors re-create a picture of themselves during the modelling process for two reasons: first, because they do not always remember the motivations nor even the conditions of their actions, second because they want to build a "nice" model rather than a true or useful one. If a model is always a creation, it should be as close as possible to reality. How an assistant agent could help an actor to stay faithful to his real behaviour? Explicit questioning, like in the air traffic control application, could be improved in taking advantage of the flexibility of a computer interface. With a simulation, it is easy to filter the perceptions of an actor, modifying actor's interface. In place of a direct question like "why did you make a new friend", which may be difficult to answer as we seen, an indirect question could be "what would you do in such situation?". For instance, in order to establish if one player's behaviour is driven by the states of his friends or the global state of the system, an assistant agent can test the reaction of the actor when some information is hidden. The agent and the actor can then better state the conditions for a given behaviour. Experimentations on similar situations than the game of Friends will allow us to test the artificial maieutic approach through perception filtering.

References

[1] P. d'Aquino, O. Barreteau, M. Etienne, S. Boissau, S. Aubert, F. Bousquet, C. Le Page & W. Dare, "The Role Playing Games in an ABM participatory modeling process: outcomes from five different experiments carried out in the last five years", in *Comm. to IEMSS*, Lugano, June 24th-27, 2002.

[2] O. Barreteau & F. Bousquet, "Role-playing games for opening the black box of multi-agent systems: method and lessons of its application to Senegal River Valley irrigated systems" in *Journal of Artificial Societies and Social Simulation*, 2001.

[3] S. Boisseau, "Co-evolution of a research question and methodological development : an exemple of companion modeling in northern Vietnam" in F. Bousquet & G. Trebuil (eds.) Companion modeling and multi-agent systems for integrated natural resources management in Asia – In Press.

[4] Candea, H. Hu, L. Iocchi, D. Nardi & M. Piaggo, "Coordination in multi-agent RoboCup teams," in *Robotic and Autonomous Systems*, vol. 36, 2001, pp. 67-86.

[5] A. Drogoul, B. Corbara & D. Fresneau, "MANTA: New experimental results on the emergence of (artificial) ant societies," in *From reaction to cognition*, lecture notes in AI n° 957, C. Castelfranchi & J.P. Müller, Ed. Berlin-Heidelberg: Springer-Verlag, 1995, pp. 13-27.

[6] Drogoul, T. Meurisse & D. Vanbergue « Multi-Agent based simulations : Where are the agents ? » in Sichman J.S., Bousquet, F., Davidsson, P., eds: *Multi-Agent Based Simulation, Third International Workshop*, Lecture Notes in Computer Science, vol. 2581, pp. 1-15, Springer, 2002.

[7] P. Guyot & A. Drogoul, , "Designing multi-agent based participatory simulations," in *Proceedings of the 5th Workshop on Agent Based Simulations*, Lisbon, 2004.

[8] R. W. Hill, J. Chen, J. Gratch, P. Rosenbloom & M. Tambe, "Intelligent Agents for the Synthetic Battlefield: A Company of Rotary-Wing Aircraft," in *Proceedings of the 9th Conference on the Innovative Applications of Artificial Intelligence*, Menlo Park, 1997, pp. 1006–1012.

[9] T. Ishida, "Q: A Scenario Description Language for Interactive Agents", in *IEEE Computer*, 2002.

[10] R. Nair, M. Tambe & S. Marsella, « Role allocation and reallocation in multi-agent teams: Towards a practical analysis », in *Proceedings of the 2nd Int. Joint conf. on agents and multi-agent* systems (AAMAS'03), 2003.

[11] M. Nguyen-Duc, V. Duong & A. Drogoul, "Agent-based modeling and experimentation for Real-time Collaborative Decision-Making in Air Traffic Management", in *Proceedings of the 24th Congress of the Int. Council of the Aeronautical Sciences* (ICAS'04), Yokohama, 2004.

[12] E. Ostrom, "Coping with the tragedy of commons", *Annual Review of Political Science*, vol 2, pp. 493-535, 1999.

[13] L. B. Said, T. Bouron & A. Drogoul, "Agent-based interaction analysis of consumer behavior," in *Proceedings of Autonomous Agents and Multiagent Systems* (AAMAS'02), Bologna, 2002.

Predictability for Autonomous Decision Support

Francisco Coelho[1] and Helder Coelho[2]

[1] Departamento de Matemática, Universidade de Évora
fc@uevora.pt
[2] Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa
hcoelho@di.fc.ul.pt

Abstract. The experimental scientist need tools to quantify and classify collected data. This paper proposes to give meaning and measure to the intuitive concept of *predictability*. It is a *global* and *time dependent* real valued quantity that, we argue, indicates *how hard it is to make a forecast for the next value* on a time series. We start with a a definition of *predictability* for binary words and show properties about its growth and computational cost. Our measure evaluates in time $\mathcal{O}\left(n^3\right)$, what is an acceptable performance specially for supporting bounded time decisions. Then, we investigate application procedures illustrated with data achieved from iterations of the logistic map, economic simulations and the Portuguese GDP (Gross Domestic Product).

1 Introduction

This research started from a simple question: *how to distinguish two agent profiles in a population, given collected samples?* While trying to answer it, we faced the need of an usable definition and measure for time-series *predictability*, the quality of being predictable.

Our frame research goal is the *design of intelligent agents* according to the vision in the *AgentLink III Roadmap* (see [1]):

> *"(...) an agent is a computer system that is capable of flexible autonomous action in dynamic, unpredictable, typically multi-agent domains (...)"*

through the *beliefs, desires and intentions* model (see [2] and [3]) augmented with

individual power the capacities for individual action and
will the processes managing the *beliefs* and *desires* to yield *intentions*;

The study of *power* remote –at least– to the work of the philosopher and logician Ingmar Pörn (see [4]). *Social Power* is mainly concerned with the *interactive, external* factors –his *roles*, *obligations*, *dependences*, etc. (see [5], [6])– of the agent's choices. With applications to the modelling and simulation of economies and, in general, normative environments, social power theoretical background is based (since [7]) on the concept of *dependence,* and is under active research since then (*e.g.* [8], [9]).

As a counterpart to social power, *individual power* is concerned with the *functional and structural, inner* factors regarding agent's choice. Research about individual power

J.S. Sichman and L. Antunes (Eds.): MABS 2005, LNAI 3891, pp. 88–98, 2006.

started (by Coelho & Coelho) in [10]. There, they exploit an economic-agent-based model, derived from [11], in order to observe and measure individual performances on agents with different action capacities, *i.e.* different individual power.

We started from that experimental setting and adapted their model to fit two alternative populations, either composed of *conformist* or of *self-promoting* agents (see section 3). The former should embody a weak form of individual power whereas the later, more ambitious, require larger capacity of action.

As a general goal, in multi-agent based simulations, the scientist should look for the effects of individual characteristics on the social dynamics (see, *e.g.* [12]). In particular, the simulation process often generate large amounts of time dependent data, *i.e. time series*. It is not always easy to extract some meaning of that data. Our simulator produced reasonable amounts of data from a selected set of observation variables. One intuitive impression from the observation of that data once plotted revealed that, for some variables, *in the conformist populations, short-term predictions seem easier than in self-promoting ones*. This conjecture obliged us to step from intuition and ask how to define and measure *predictability*.

We propose a meaning and a measure to *predictability* of time series. This measure should be a global, time-dependent quantity expressing *how hard –or easy– it is to make a forecast for the immediate future*.

It turns out that our measure candidate evaluates reasonably fast. This suggests an unexpected application: Its use by software agents for bounded-time decisions. Consider the case of an agent facing a decision problem. The predictability of each entity involved in that problem, being quickly computable from historical data, can become a valuable factor on its decision. Thus,

> *we are proposing a measure of predictability that, because of its fast computation, can be used by bounded-resource autonomous agents.*

Most usual statistical quantities, *e.g.* mean and variation, although providing global information, are time independent: changing the order of terms in a time series doesn't affect its mean value. Instead, in the intuitive notion of prediction, time *does* matter: one would change the forecast on permuted data. There are sophisticated statistical procedures and measures to deal with time series (see [13]), namely *stationarity*, *ergodicity*, the *auto-correlation, auto-covariance* functions and others. Besides, it is usual to apply *Fourier Transforms* to study the resulting series in *frequency-domain* instead of the original time series in *time-domain*. The *ARIMA* technique, due to its complexity, is not of easy application (yet) so cannot be considered feasible by autonomous agents. Hidden-Markov-models (HMM's) [14]require a model (of the transitions and output probabilities) or use the Baum-Welsh algorithm to estimate it. Furthermore, HMMs require predicted symbols to compute the most likely sequence of hidden states, by the Viterbi algorithm [15]. Resuming, HMMs, although feasible, do not provide a direct answer to our quest: A fast algorithm to measure predictability of time-series. We believe that there is room for a simpler approach.

Another road to predictability is through the *computational complexity* required to reconstruct the observed word. Let $s = s_1 \cdots s_t$ be a binary word. Slightly incorrectly, say that the *Kolmogorov complexity* [16] of s is the minimal number of states required in a *finite state automaton* (FSA) to reconstruct s. A (binary) FSA [17] is defined by a *set of states* $N = \{0, \dots, n\}$, a *state transition function* $\delta : N \times \{0,1\} \to N$ and the *subset of accepting states* $F \subseteq N$. Given the (binary) input $s = s_1 \cdots s_t$, the final state σ_t is recursively defined by $\sigma_t = \delta(\sigma_{t-1}, s_t)$ and $\sigma_0 = 0$. If $\sigma_t \in F$ then s is *accepted* or *recognized*. Write $A(s) = 1$ if A accepts s and $A(s) = 0$ otherwise. FSA can be used to (re)construct words: Given a FSA A and $s = s_1 \cdots s_t$, define $s^{(1)} = sA(s)$ by the *concatenation* of the symbol $A(s)$ to the end of the word s. Denote by $S_k(w)$ the w suffix (see below) of length k and repeat:

$$s^{(i+1)} = s^{(i)} A\left(S_k\left(s^{(i)}\right)\right).$$

Say that the resulting word after m iterations is *reconstructed* by A: One need only a seed with size k (the suffix $S_k\left(s^{(i)}\right)$) and A to get the reconstructed word. In other words: *the future of s can be foreseen knowing A and k consecutive values on the past.* The more states required in a FSA to perform this prediction, the more *unpredictable* (in the sense *hard to predict*) the time series is.

The problem with this approach is that there are n^{2n} binary FSA with n states, rendering unfeasible a brute force search –the best known– for a minimal FSA reconstructing a given word.

We are taking a different, combinatorial like, approach. In our inital setting it is given a binary word $s = s_1 \cdots s_t$ with t symbols chosen from $\mathbf{2} = \{0, 1\}$. The goal is to measure *how hard it is to make a forecast for the next value* s_{t+1} and not actually predicting it.

A naive guessing procedure would consider the present observation s_t and look into the past, to find what happens when that value was observed, say $s_\tau = s_t$ with $\tau < t$, and propose that $s_{t+1} = s_{\tau+1}$. It is easy to see that this strategy will plainly fail almost always. However, it can be improved if, instead of taking into account only the last observed value, one look for previous occurrences of the *recent past*, say $v = s_{t-k} \cdots s_t$, such that $s = s_1 \cdots s_{t-k-1} v$.

Suppose that this extended method works with some word s. For a given value of k, whenever the sub-word v occurs in s, the next value is always equal. This is a statement about the *predictability* of s: To predict the future s_{t+1} it is reasonable to look for the sub-word of the last k observations v and expect that the immediate future is the value that *always* has occurred after that sub-word. Notice that in particular this procedure *does* work with (discretizations of) any periodic function, for example $\sin$ and $\cos$.

Inversely, if the word v occurs only once, then s_{t+1} can't be predicted –this way– from $s_{t-k} \cdots s_t$. We will relate the (un)predictability of s with the size required to define this solitary suffix. The leading intuition here is:

- The rarer the present, the harder it is to forecast the future and
- the more common the recent history, the longer the history we must consider to make it unique.

As an illustration, consider the word with 5 constant observations $c = 11111$. It is "very much" predictable: the present observation c_5 is equal to any other before, i.e. it is "very much" non-rare. To make a "recent past" segment $c_{5-k} \cdots c_5$ solitary in the entire word c one must take $k = 4$ since any shorter segment occurs at least twice in c.

On the other hand, the word with a sudden change in the end $e = 00001$ is "very little" predictable: the last observation $e_5 = 1$ never occurred before, i.e. it is "very little" non-rare, so we are hopeless to forecast what shall come next. The suffix e_5 occurs only once in the entire word e what is enough to make the present moment solitary.

2 Predictability

Let s be a binary word with n symbols, $s \in \mathbf{2}^n$. A word w is a *sub-word of* s if there exists words x and y such that $s = xwy$. If $y = \varepsilon$ (the empty word) say that w is a *suffix* of s and if $x = \varepsilon$, say that w is a *prefix* of s. Denote the *number of symbols in* s by $|s|$ and the *number of occurrences of the sub-word* w *in* s by $|s|_w$. Now, define

$$\nu(s) = |w|$$

where w is the smallest suffix of s such that $|s|_w = 1$. We say that a suffix w is *unique* if the suffix sub-word w occurs only once in (the end of) s. Now, the *predictability* function σ is defined by

$$\sigma(s) = \frac{\nu(s)}{|s|}.$$

So, $\sigma(s)$ is the ratio of the length of the shortest unique suffix of s in s.

2.1 Properties

In order to illustrate the predictability function, consider a few words of size 6:

	word	singular suffix	σ
Constant	111111	111111	1.0
Parity	010101	10101	$5/6 \approx 0.84$
Step	000111	111	0.5
Signal	000100	100	0.5
Jump	011110	10	0.33
End	000001	1	$1/6 \approx 0.17$
Random1	101101	1101	0.67
Random2	010110	110	0.5

Notice that the Random1 word has (relatively) high predictability value, reflecting is internal repetitive shape.

The predictability function has a few easy properties: Let $s \in \mathbf{2}^n$, $\bar{s}$ the *complement* of s, the word obtained from s by $\bar{0} = 1$ and $\bar{1} = 0$. Then, $\sigma(s) = \sigma(\bar{s})$; $\sigma(s) = 1$ iff s or $\bar{s}$ is the Constant and $\sigma(s) = 1/n$ iff s or $\bar{s}$ is the End; $1/n = \sigma(\text{End}) \leq \sigma(s) \leq \sigma(\text{Constant}) = 1$.

And, less easy, for any digit $D \in \mathbf{2}$,

If $\nu(sD) > \nu(s)$ *then* $\nu(sD) = 1 + \nu(s)$.

Proof. Suppose that $s = xw$ and that w is the smallest suffix of s that occurs only once in s. Then $\nu(s) = |w|$. Now $sD = xwD$ and let $\nu(sD) = |w'|$, where $sD = x'w'$

and w' is the smallest suffix of sD that occurs only once in sD. Since, by hypothesis, $|w'| > |w|$ then $w' = ywD$ for some, possibly empty, word y. Notice that different occurrences of ywD in sD imply different occurrences of wD in sD and that imply different occurrences of w in s. But there is exactly one such occurrence. Therefore wD is the smallest suffix of sD that occurs only once in sD, so $\nu(sD) = |wD| = 1 + |w| = 1 + \nu(s)$. QED.

This property can be stated as *the light speed limit* or *the memory of the past*: time series can't quickly improve its predictive quality.

The computational cost of ν and predictability evaluation, at least compared with Kolmogorov complexity, is very acceptable: given a word s of size $|s| = n$, in the worst case (constant) it is required to consider the n suffixes $w_1 = s_n, w_2 = s_{n-1}s_n, \ldots, w_n = s_1 \cdots s_n$. Each suffix candidate, say $w_k = s_{n-k+1} \cdots s_n$, is successively compared with each one of the $n-k+1$ lenght k sub-words $s_1 \cdots s_k, s_2 \cdots s_{k+1}, s_{n-k+1} \cdots s_n$. At last, each of these comparisons involves doing k tests of symbol equality. Thus, to compute ν there are required

$$\sum_{k=1}^{n} k\,(n-k+1) = \frac{1}{6}\left(n^3 + 3n^2 + 2n\right)$$

tests of symbol equality.

2.2 Real Valued Series

Consider, for example, two plots of the logistic equation $x_{n+1} = a x_n (1 - x_n)$. The first plot, depicted in Figure 1, for $a = 3.741$, shows a periodic orbit whereas the second plot, with $a = 3.98$, is in the chaotic regime. Right panes of both figures show the succesive σ values. Remark that in the periodic regime predictability approaches 1 whereas in the chaotic regime predictability falls near 0.

The fact that the series values are real values and not binary values is easy to overcome: one can always take the binary representation of the rounded values, up to some fixed precision. The result is a binary word $w = X_1 \cdots X_t$ where each X_i is a binary word of length, say, k. But a more subtle issue arise: For the ν and σ computations, it makes no sense to consider *subwords "in-between" values*: If x_i, x_{i+1} are consecutive terms in the real valued time series, digitized into the word $\cdots X_i X_{i+1} \cdots$ then, for the

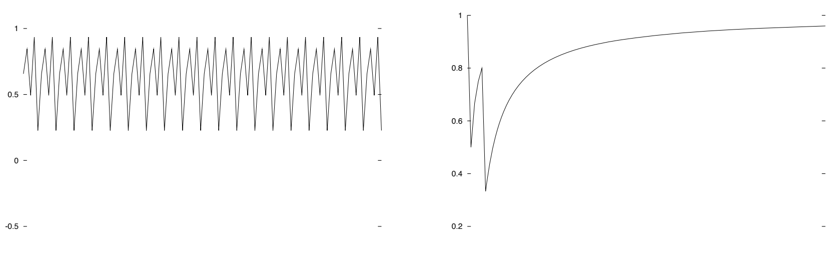


Fig. 1. The logistic equation on a periodic regime ($a = 3.741$) and the σ evolution

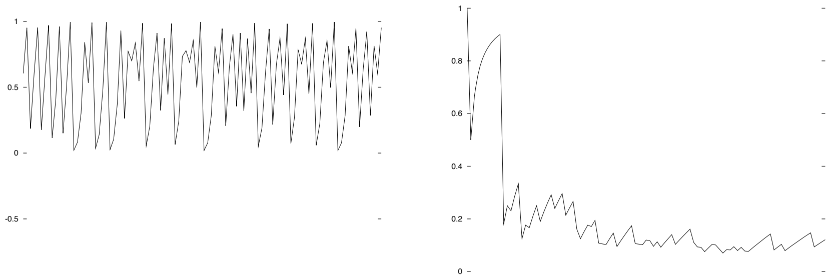


Fig. 2. The logistic equation on a chaotic regime ($a = 3.98$) and the σ evolution

ν computation, while searching the there would be accounted subwords $w = xy$ where x is a suffix of X_i (i.e. $X_i = ax$) and y is a prefix of X_{i+1} ($X_{i+1} = yb$), but w has no relation at all with the x_i, x_{i+1} values.

To workaround this issue one can consider, instead of length 1 binary symbols, length k words (k is the number of bits used to code each real number) and proceed as before: Suppose that $s \in \mathbf{2}^{kt}$. Denote by $|s|^{\langle k \rangle} = t$ and, if $w \in \mathbf{2}^{kn}$, $|s|_w^{\langle k \rangle}$ is the number of occurrences of w in s, *starting at positions* $\equiv 1 \,(\mathrm{mod}\, k)$. Now, define $\nu^{\langle k \rangle}(s) = |w|^{\langle k \rangle}$ where w is the smallest suffix of s such that $|s|_w^{\langle k \rangle} = 1$ and the *k-predictability* function

$$\sigma^{\langle k \rangle}(s) = \frac{\nu^{\langle k \rangle}(s)}{|s|^{\langle k \rangle}}.$$

The stated properties for $k = 1$ carry without trouble to this general case. Remark that for even values of k the Parity is “constant”, and, as such, $\sigma^{\langle 2p \rangle}(\text{Parity}) = 1$. We will use the lighter notation σ instead of $\sigma^{\langle k \rangle}$ since it is clear when we are dealing with real valued series and in those cases we always used the value $k = 16$.

3 Application

As stated before, this work was motivated by the study of the effects of individual profiles in the social dynamics, through the analysis of an economic multi-agent based simulation. We used an hierarchical tribute model introduced by Caldas [11] and further investigated by Coelho & Coelho [10]. A complete description of this model environment, agents and dynamics falls off the scope of this paper, so we refer the details to the papers where it is first described and explored.

Briefly, in this model, agents are engaged in the production of a single good. They are gathered in hierarchical production groups, whose structure defines individual revenue and contribution. Below a survival threshold agents are replaced, not necessary on the original group.

We changed the original model to a system where the dynamical rules allow agents to choose future actions. Here, we aim the study the social effects if agents have different action strategies, in our framework simplified to the choice of group migration/foundation vs. continuation on the same group. To that end we selected a set of variables measuring individual and social quantities: among others, *individual age*, *social rank*, *productivity* and *group age*, *size*, *depth*, *leadership (size)*.

Those societies where then populated with two profiles for individual agents: The *conformist* profile, where agents stay on the original group until eventually die and the *self-promoting* profile, where agents try to reach leadership positions. For the conformist profile we used the (passive) agent control from [11] but the self-promoting behaviour was created for this experiment.

Self-promoting agents take into account their social rank (defined in [10]) and productivity, used as a measure of wealth. If the (individual) productivity surpasses a given threshold and the agent is below a given social rank, then he deserts from his group, randomly choosing either to join another group or found a new one.

Remark that in this experiment all agents exist in equally ruled environments, being all differences located at the individual (agent) action selection level.

The next study step was to run several instances of those societies, collecting the run values of the selected variables. That resulted on a set of data for subsequent statistical analysis. One of those variables is the *Average Group Leadership Size* computed, at each time-step, by the mean value of the number of agents placed in top hierarchical places of all the production groups. While in conformist societies this variable is almost constant, in self-promoting societies, as expected, its dynamics is more complex.

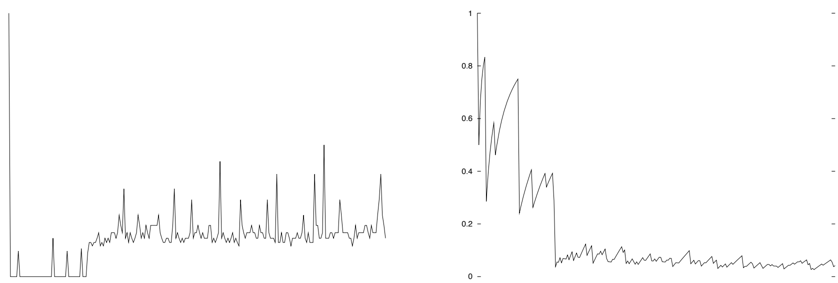


Fig. 3. *Average Group Leadership* in a *self-promoting* sample and the σ evolution

Since the data relative to the conformist sample is, after the transient period, constant, we omit its display. However, with respect to the self-promoting sample, this variable has a much more interesting behavior. The left side of Figure 3 shows time evolution of *Average Group Leadership Size* in a self-promoting sample and suggests that

> *while forecasting is easy in the conformist sample, it is much harder in the self-promoting one.*

We apply the predictability function in order to provide a quantitative measure to support this statement. Consider the right side of Figure 3. While in the *conformist* samples predictability is almost constant, near 1.0, for the *self-promoting* data it decreases to near 0 values, *i.e.* for these samples,

> *forecasting is possible in the conformist profile what is not the case in the self-promoting sample.*

Now one can process all the collected samples and give statistical soundness to the general statement

> *Group Leadership Size* is predictable in *conformist* societies and not predictable in *self-promoting* societies.

Table 1 resumes our collection of 40 conformist and 40 self-promoting samples.

Table 1. Predictability conclusion about *Group Leadership*

Group Leadership	average	variation
conformist	**0.99599**	0.00000
self-promoting	**0.09606**	0.04547

The procedure for the *Group Leadership* table can now be repeated for every variable and from the predictability information of each variable we can build an overall picture of the predictability on the two profiles.

More precisely, given sets $X = \{x_1, \ldots, x_n\}$ of variables and $S = \{s_1, \ldots, s_m\}$ of samples, let $x_i(s_j)$ be the time series defined by the observation of variable x_i on sample s_j and

$$\sigma_{ij} = \sigma\left(x_i\left(s_j\right)\right)$$

the respective predictability value. Now, if $S' \subseteq S$, let

$m_i(S')$ be the mean of $\{\sigma_{ij} : s_j \in S'\}$,
$v_i(S')$ the respective variation,
$M(S')$ the mean of $\{m_i(S') : x_i \in X\}$ and
$V(S')$ the mean of $\{v_i(S') : x_i \in X\}$.

Table 2 shows the M and V values when X is the set of all recorded variables and S' for the first row is the set of conformist samples and for the second row is the self-promoting sample. This table shows evidence that

> *Self-promoting societies are less predictable than Conformist societies.*

Table 2. Overall statistics of predictability

All variables	M	V
conformist	**0.30410**	0.00002
self-promoting	**0.06537**	0.03428

Table 3. Predictability in individuals vs. groups

Groups vs. *Individuals*	indiv. M	indiv. V	group M	group V
conf.	**0.15808**	0.00003	**0.40631**	0.00001
self-p.	**0.06155**	0.03193	**0.06805**	0.03592

If we manage the set of variables X further information can be extracted. In table 3 the first two columns are relative to the *individual variables* (*individual age, individual rank, individual productivity, etc*) and the third and forth columns are relative to *group variables* (*group size, group depth, group capacity, etc*).

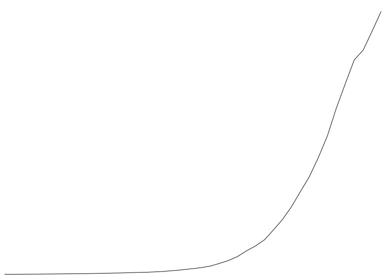
We claim that these numbers illustrate two achievements in our work. First, that *will* and *individual power*, here simplified to the caricature form of choice to desert from a group, affect the social dynamics. This is clear from the statistics of the predictability function on the two populations.

Second, the distinction of the two samples profiles places the predictability function as a potential candidate to the study of dynamical systems.

4 Conclusion

We have defined a function σ providing a measure of time series that is both *global* and *time dependent*, as should be any measure of *predictability*. More, the predictability of long enough periodic time-series evaluate to near 1.0 values and, although defined for the present (or most recent) state, the σ function accounts for the effects of past occurrences. With the computation time bounded to $\mathcal{O}\left(n^3\right)$, this function provides a sensible tool for autonomous decision support.

Moreover, predictability might be used as an analysis tool. We illustrated a possible use is section 3. Predictability successfully distinguishes the *conformist* from the *self-promoting* samples.



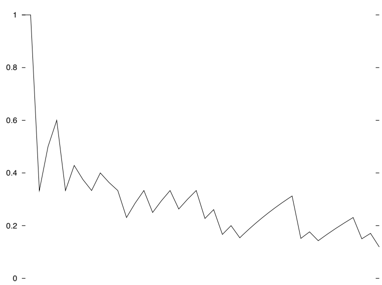


Fig. 4. The Portuguese GDP and the σ evolution

The predictability function doesn't perform well on exponential data. As an example, consider the Portuguese GDP from 1954 to 1995, depicted in the left pane of Figure 4. Although its clear trend, the σ plot (in the right pane) approaches 0. For linear (in general, for polynomial) data $x = \{x_1, \ldots, x_n\}$ one might study the finite differences $d_x = \{x_2 - x_1, \ldots, x_n - x_{n-1}\}$ but for exponential data this time-series remain exponential.

5 Future Work

There are many paths to further research the σ function and the *smallest unique suffix*:

- Allowing some sort of fuzziness on the sub-words;
- Classifying time-series through their predictability evolution;

- Investigating causality relations and next symbol probability distributions concerning the smallest unique suffix;
- Simulating agents taking decisions based on predictability;
- Relating σ and the Lempel and Ziv algorithm;
- others...

In particular, we are interested in two lines:

Fuzziness on sub-words. The predictability function settles on *exact* matching of binary words but for real valued data it is reasonable to deal with *approximations*. Consider a real valued time series $s = s_1, \ldots, s_n$ and a suffix $w = w_1, \ldots, w_k = s_{n-k+1}, \ldots, s_n$. Say that w *matches* $y = y_1, \ldots, y_k$ if $\|w - y\| < \varepsilon$ for some fixed ε and k dimensional norm $\|\cdot\| : \mathbf{R}^k \to \mathbf{R}$ (e.g. $\|v_1, \ldots, v_k\| = \sqrt{\sum_{i=1}^{k} v_i^2}$). Now, define

$$\nu^{\langle \varepsilon \rangle}(s) = k$$

where k is the dimension of the smallest suffix w of s such that the number of matches of w with k dimensional "sub-words" of s is 1 and the *predictability* function

$$\sigma^{\langle \varepsilon \rangle}(s) = \frac{\nu^{\langle \varepsilon \rangle}(s)}{n}.$$

It is not clear yet how does $\nu^{\langle \varepsilon \rangle}$ and $\sigma^{\langle \varepsilon \rangle}$ relate with the $\nu^{\langle k \rangle}$ and $\sigma^{\langle k \rangle}$ used before.

The $\nu^{\langle \cdot \rangle}$ and $\sigma^{\langle \cdot \rangle}$ functions can also be applied to multi-dimensional data. One possible way of doing that is through the $\mathbf{R}^k \times \mathbf{R}^n \equiv \mathbf{R}^{k+n}$ isomorphism. Multi-dimensional time series are particularly interesting due to the *Takens Embedding Theorem* [18] that provides key information about the dynamics of the observed system and, perhaps, cope the exponential problem exposed before.

Decision support. Besides these mathematical developments, required to settle, expand and refine the applicability of the σ function, we account to use it in our quest for *individual power design*. Not only as a tool to classify behaviour profiles, in the lines of this paper, but by performing an experimental set to inquiry the individual power performance of agents aware of predictability. More precisely, we will explore how software agents can use predictability for decision support and how does that use relates with individual power and will.

Acknowledgements

We would like to thank the financial support from the Portuguese agency *Fundação da Ciência e Tecnologia* through the R&D unit LabMAg and the project MAGO2.

References

1. Luck, M., McBurney, P., Shehory, O., Willmott, S., the AgentLink Community: Agent technology: Computing as interaction –a roadmap for agent based computing–. Technical report, AgentLink (2005)
2. Wooldridge, M.: An Introduction to MultiAgent Systems. John Wiley and Sons Ltd, Chichester (2002)

3. Walczak, A.: Planning and the belief-desire-intention model of agency. Technical report, University of Hamburg (2005)
4. Pörn, I.: The Logic of Power. Basil Blackwell, Oxford (1970)
5. Fasli, M.: On the interplay of roles and power. In Skowron, A., Barthes, J.P., Ron Sun, L.J., eds.: IEEE/WIC/ACM International Conference on Intelligent Agent Technology. (2005) 499–503
6. López y López, F., Luck, M., d'Inverno, M.: Normative agent reasoning in dynamic societies. In: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS). (2003)
7. Castelfranchi, C.: Social Power: A point missed in Multi-Agent, DAI and HCI. In Demazeau, Y., Müller, J.P., eds.: Decentralized A.I., North-Holland (1989) 49–62
8. Castelfranchi, C., Falcone, R.: Principles of trust for MAS: Cognitive anatomy, social importance and quantification. IEEE (1998) 72–79
9. Castelfranchi, C.: The micro-macro constitution of power. personal communication (2003)
10. Coelho, F., Coelho, H.: Towards individual power design. In Pires, F.M., Abreu, S., eds.: Progress in Artificial Intelligence. Volume 2902 of Lecture Notes on AI., EPIA 2003, Springer (2003) 366–378
11. Caldas, J.C., Coelho, H.: The interplay of power on the horizon. In: Proceedings of the II International Conference on Computer Simulation and the Social Sciences, Paris (2000)
12. Grimm, V., Railsback, S.F.: Individual-based Modelling and Ecology. Princeton University Press (2005)
13. Shalizi, C.R.: Methods and techniques of complex systems science: An overview. (2003)
14. Fraser, A.M., Dimitriadis, A.: Forecasting probability densities by using hidden markov models with mixed states. In: Time Series Prediction: Forecasting the Future and Understanding the Past. Addison Wesley (1993)
15. Viterbi., A.J.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory **13** (1967) 260–267
16. Vitanyi, P.: Algorithmic chaos. (2003)
17. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata and Formal Languages. Addison-Wesley, Reading, MA (1979)
18. Takens, F.: Detecting strange attractors in fluid turbulence. In Rand, D.A., Young, L.S., eds.: Symposium on Dynamical Systems and Turbulence, Berlin, Springer-Verlag (1981) 366–381

Analyzing Police Patrol Routes by Simulating the Physical Reorganization of Agents

Adriano Melo, Mairon Belchior, and Vasco Furtado

UNIFOR – University of Fortaleza, Master of Informatics (MIA),
Av. Washington Soares 1321, Edson Queiroz,
Fortaleza – Brazil
vasco@unifor.br, aanmelo@hotmail.com, maironb@gmail.com

Abstract. In this article we describe a tool for assisting the investigation of different strategies for the physical reorganization of agents. We show how the tool was used in the public safety domain to help in the study of strategies of preventive policing. A society of agents that simulates criminal and police behavior in a geographical region was constructed. In this society, artificial agents representing the police are responsible for preventing crimes. The organizational structure of the police is characterized by the existence of a centralized command that has the task of distributing and redistributing the police force in a region according to an analysis on crime and the factors that influence it. The simulation of different strategies of physical reorganization is a first step to better understand the influence that different police patrol routes have on the reduction of crime rates.

1 Introduction

Multi-Agent Systems (MAS) must allow dynamic adaptation of organizations as changes occur in the environment. Typically, reorganizations are carried out by external intervention of a programmer, but for an MAS to be truly autonomous, mechanisms for dynamic reorganization must be available. The concept of *dynamic adaptation* refers to modifying the structure and behavior of an MAS, such as adding, removing, or substituting components, done while the system is running [13]. Dynamic adaptation demands that systems evaluate their own state and take action to preserve or recover it, by performing suitable integration and reconfiguration actions. Most existing approaches to reorganization consider only behavioral aspects affecting the agents [4] [9]. Recently [7], in addition to behavioral aspects, some approaches have proposed also considering situations in which the social structure of the society changes.

Our research work concentrates on a specific way of reorganization we call physical reorganization. This kind of reorganization is found in artificial agent societies in which the static or dynamic physical position of the agents is represented. In particular, such a representation is very important in geosimulation systems. Geosimulation is an urban phenomena simulation model that uses the multi-agent methodology to simulate discrete, dynamic, and event-oriented systems [3].

One of the crucial questions regarding the control of crime and violence in urban centers is how to gauge the actual impact of certain police management strategies

J.S. Sichman and L. Antunes (Eds.): MABS 2005, LNAI 3891, pp. 99 – 114, 2006.

upon the regulation of crime rates. This is indeed a difficult question to be answered, as it seems that the effectiveness of a certain public-safety policy on a given metropolitan region depends, directly or indirectly, upon an array of factors, ranging from the levels of concentration of wealth to the physical organization of the urban center under consideration. In such a context, it is quite consensual that police patrolling can be considered as one of the best known means for implementing preventive strategies towards the fight against crime.

In this article we describe a tool for assisting the investigation of different strategies of physical reorganizations. This tool has already been used to help in the understanding of behavioral and structural dynamic reorganization of agent societies [8]. Here we focus on how this tool can help in studying physical reorganization of agent societies. We show how the tool has been used in the public safety domain for helping in the study of strategies of preventive policing. A society of agents which simulates criminal and police behavior in a geographical region was constructed. In this society, artificial agents representing the police are responsible for preventing crimes. The organizational structure of the police is characterized by the existence of a centralized command that has the task of distributing and redistributing the police force in a region according to an analysis on crime and the factors that influence it. The simulation of different strategies of physical reorganization makes it feasible to better understand the influence that different police patrol routes have on the reduction of crime rates.

This article is structured as follows. Initially, we describe related works and present the problem domain. Next we describe the architecture of a simulation tool designed for use as a tool for analysis of different agent reorganization strategies. With the simulation, we aim at analyzing and comparing the effect of different reorganization strategies. In particular, we want to understand the effects of different dynamic physical reorganization strategies on criminal activity. We then describe some examples of the use of this tool and also describe some qualitative and quantitative results we obtained with its use.

2 Related Works

In order for an MAS to behave productively (i.e., coherently), some sort of coordination policies, protocols, and mechanisms must be properly configured and deployed. One such coordination mechanism comes in the format of organizations. Briefly speaking, an organizational structure can be simply perceived as a set of mutual restrictions adopted by a group of agents so that they can more easily achieve their local objectives towards the accomplishment of the group's overall objectives [11]. Despite its positive aspects, designing an effective multi-agent organization tailored to the peculiarities of a given application scenario is not a trivial task, mainly when the domain of study is like the one considered in this work, namely, multi-agent patrolling. Despite its potentially wide-ranging applicability, just a few studies have been conducted on the theme of multi-agent patrolling. One justification for such a fact is that existing approaches to dealing with some related problems, such as the traveling salesman problem–TSP [12], cannot be directly applied, or even adapted, for coping with the intricacies of the patrolling task.

One prominent research work in such context was recently developed by [1], having as basic motivation to provide answers to the following questions: Which kind of MAS architecture should be selected by the designer for tackling a given patrolling task? What are the means to properly evaluate an implemented MAS dedicated to patrolling? To what extent do parameters like size and connectivity influence the overall MAS performance? In such regard, different MAS architectures have been conceived and evaluated experimentally by the authors, making it possible to elicit some preliminary guidelines for the suitable design of MAS for patrolling. The devised methodology involves both the identification of some evaluation criteria and the definition of some dimensions of characterization of the MAS architectures.

Regarding the first of the above issues, each patrol agent tries to maximize the number of visits in the places to be patrolled in order to reduce their global “idleness”. The idleness of a place refers to the average period of time elapsed between two consecutive visits of at least one member of the police patrol force. The worst node idleness value (among all the places) and the time necessary for all the police troops to visit all the patrol points, at least once, are then taken into consideration as evaluation criteria for assessing different patrolling strategies. Regarding the dimensions of characterization of a possible multi-agent patrol architecture, the authors have examined the following ones: agent type (reactive vs. cognitive); agent communication (allowed vs. forbidden); coordination scheme (central and explicit vs. emergent); agent perception (local vs. global), and decision-making (random selection vs. goal-oriented).

Following another direction, Winoto [14] has made use of the multi-agent paradigm for representing and characterizing some important crime features. In this work, an economic perspective upon crime is elaborated and the notion of impunity, which seems to be an essential factor for the increase/decrease of crime rates, is analyzed from the viewpoint of crime repression. The preventive aspect, however, is somewhat neglected by the author.

In our earlier work, we modeled the typical profiles of criminals and police officers in terms of artificial agents in order to develop an intelligent tutorial system [10]. The ExpertCop system comprehends a full-fledged geo-simulation environment focused on the analysis of criminal activity, which was conceived to support police managers in learning, through interactivity, how to properly allocate, on a given geographical map, the human resources currently available.

Despite their innovative ideas, all of the above-mentioned approaches do not systematically investigate one important issue underlying the multi-agent patrolling task: How to devise alternative ways leading to the design of police patrol routes in consonance with the peculiarities of the patrolling scenario under consideration.

3 The Public Safety Domain and the Police Allocation Task

The allocation of police-officer resources in urban areas in order to perform preventive policing is one of the most important tactical management activities for controlling criminal activity, which is usually decentralized among sub-sectors of the police departments covering different zones or neighborhoods of the territory. What it is intended from those tactical managers is that they periodically analyze the disposition of crime in their region and then perform the (re)allocation of the police force based on such analysis.

An underlying hypothesis of such allocation work is that, by knowing where the crime is currently happening and its associated reasons, it is possible to make a more optimized distribution of human resources and, consequently, to decrease the overall crime rate. However, the high volumes of information that police departments have to analyze is one of the main difficulties to provide society with effective answers. Tactical managers that perform police allocations, for instance, have difficulties in untangling the complex relationships between the different factors that influence the several types of crime occurrence.

In fact, understanding criminal mapping activities, even using geographic information systems, is a non-trivial task. In addition to that, real-life experiments in this domain cannot be performed without high risks, as they may result in loss of human lives. In this context, simulation systems for decision support come to be a prominent tool. Following this point of view, in this work we concentrate on the description of one such simulation-based tool focused on the investigation of alternative configurations (i.e., physical disposition) for police patrolling in an artificial environment that mimics a certain demographic region.

The conceptual basis for preventive approaches and the development of some pro-active policing strategies can be found in "*Routine Activities Theory*" Cohen and Felson [5], which attempts to explain the evolution of crime rates not only through the characteristics (psychological profiles) of the offenders, but also through the circumstances in which crimes occur. Basically, Cohen and Felson [5] point out that, in order for a criminal act to take place, three elements must coexist: a motivated offender; a suitable target, either an object or person that can be attacked; and the absence of capable guardians in charge of the preventive actions. The crime model derived by the authors is then based on an economic equation involving the aforementioned elements. A direct conclusion drawn from such work is that criminal offenses are related to the nature of everyday patterns of social interaction. Another is that the police force is, naturally, the central element for promoting public safety by means of diligence and dissuasion. Basically the goal of the police is to reduce or at least to keep crime under control. The main variable that police force has under control is the physical disposition of the patrol, which can be a static position or routes. Therefore, a tool for helping in the investigation of alternative configurations of police patrol in an artificial environment is very welcomed to the police. In this context, issues like the size of police patrol routes, the moment to reorganize these routes, and the triggers that motivate such reorganization, are still open.

3.1 The Agent Society

The agents that are part of the society are the following:

- Notable points: They are the commercial or entertainment establishments in the area such as drugstores, banks, gas stations, lottery houses, squares, and shopping centers.
- Police: Their function is to avoid the occurrence of crimes. Each police team should have at least one route[1], and with this route they will be accomplishing the preventive policing of the area they occupy.

[1] Routes are a set of points that police officer must go through during a determined period of time at a defined speed.

- Criminal: This is the person who executes the crimes. All criminals possess a vision that allows them to see cells (the environment is represented by a grid of cells) around them. They can see around them according to the value of the vision and the size of each cell. For example, with a vision of 1000 meters, if each cell has 100 meters, the criminal will be able to see 10 square cells around him.

There are two objects that are part of the simulation, but they are not characterized as agents. The police stations represent the police officers' initial point from where they proceed/arrive to/from their routes. The criminals' residence is the point where the criminals should be during the period that they are not committing crimes.

3.2 Criminal Agent Behavior

Each criminal has four actions: to choose a target, to move, to evaluate whether a crime should or should not be committed, and to commit (or not) a crime. Criminal motivation is driven by their goals. The criminal's selection of the targets or objectives is made according to an estimate of the distribution of crimes in an area. We have characterized crime data over a 2-year period in the State of Ceará in order to estimate the probability of occurrence certain crimes in a specific police patrol area. In our studies, we have focused on six types of crimes against the property. Each one of these crime types has its corresponding target as those shown in Table 1.

Table 1. Probability of targeting

Prob.	Target type
10%	Square
15%	Drugstore
10%	Lottery House
15%	Gas Station
20%	Shopping Center
30%	Bank

After their objective is defined, all criminals ask the environment the route to the closest exemplar of the notable point selected as the objective. The time expended to reach to goal is calculated based on the distance to the target and the speed of the criminal, which, in our studies, is constant and the same for all the criminals. The shortest time is taken as a basis so all others move only during this time. In each tick of the simulation, a criminal moves in the direction of the selected target. Finally, the decision whether or not to commit a crime is made.

For the criminal decision, the following factors are analyzed in that sequence: the probability of a crime being successfully or unsuccessfully effectuated and the existence of police within the field of the criminal's vision at the moment. Such a probability involves the other aspects, besides police presence, that influence the criminal decision, such as target vulnerability, target value, escape route, and demographic density. All criminals have a probability of deciding to commit a crime that is based on their analysis of being successful or unsuccessful. This probability is defined from the criminal's life history. The experience η is measured by an expression that takes into account the committed crimes and the unsuccessful crimes as described bellow.

$$\eta = (NbU\ /TNbt\ )\times\left(e^{-(NbS\ /\, TNbt\ )}\right)$$

Where *NbU* is the number of unsuccessful crimes, *TNbt* is the total number of crimes attempted by the criminal and *NbS* is the number of successful crimes.

This expression aims at capturing the experience level of a criminal. The idea is represent the fact that the more a criminal commits crimes, the greater the probability of being successful. This function also aims at representing the punishment that a criminal receives when he decides to commit a crime and such decision is unsuccessful. The probability of deciding to commit a crime is then a trade-off between the unsuccessful and successful experiences. The unsuccessful experiences have a greater weight because they represent the time of inactivity of the criminal after being caught.

There are two possible results for the function of decision: the first one is the criminal does not commit the crime, and will then select a new objective. The second result is to decide to commit the crime. In that case, if the criminal is within grasp of a police team in the region, then the crime is considered unsuccessful; otherwise a crime is committed.

3.3 Society Organization

In the society described above, it is only relevant to study the organization of the police[2]. The basic organization of the society of police agents is eminently hierarchical. This organization follows a military structure where ranks determine the degree of authority. For the purpose of this work, we opted to represent a simple hierarchy with only one level of command. A colonel has the responsibility of defining patrol routes for a certain area of the city. Each route possesses a police team that may be composed of one or more police officers. The organizational structure is thus hierarchical and the autonomy for reorganization only exists at the central level. We point out that our goal at the present is not to study reorganization with different strategies of coordination as a decentralized one. What we intend is to capture essential notions relative to reorganization strategies that will eventually be useful to implement different coordination schemes.

4 The Reorganization Tool

We have developed a simulation tool for studying different reorganization strategies in MAS. It is an extension of Repast developed at the University of Chicago [6].

The tool contemplates three types of reorganization: behavioral, structural, and physical. Each reorganization type can be configured in independent ways. Users can choose which one or which ones to use in their simulation. For each reorganization,

---

[2] Although we may think about modeling criminal organizations, for the purpose of this work, we have concentrated on the police organization.

the behavior of the agent when making decisions concerning reorganization is represented in production rules (conditions and executions).

In Figure 1 we show one screen shot of two windows for the physical reorganization of the tool. The one on the left shows an example of how a behavioral rule of an agent can be built with the help of the tool. The condition of the rule can be a comparison between variables or a comparison of variables with a certain value, and could be a simple or composed statement. After the creation of the condition, the action that will be executed, if the rule is satisfied, should be included. It is important to highlight that the variables that compose the condition and actions are the properties and methods defined in the agent society. They are supplied by the simulation generalizing the reorganization tool so that it can be used by any domain. The window on the right side shows all the parameters that are used by the domain of public safety, such as the number of police teams, the number of criminals, the criminal's vision, among others.

Originally, each type of reorganization can be simulated through two reorganization strategies:

- The centralized strategy (role-based), assumes the existence of an agent who can have access to all of the information on the other agents at any moment, and may thus decide to provoke reorganization in the agent society.
- The shared strategy – all of the agents or a part of them will be performing the role of the centralized agent, however with no autonomy to execute the changes without beforehand putting them to a vote of the other agents that are part of the society.

At that moment in our studies, we focus on the centralized strategy. The tool also allows that the simulation to be made without any possibility of reorganization, and thus not considering any of the rules created by the user.

The simulation tool provides different manners for evaluating of the quality of the reorganization strategies. It makes it possible to store the results at the end of the simulation, and the data that was previously defined by the user as points of evaluation of his/her simulation can be accessed. Thus it will be possible to generate comparative graphics in several formats.

5 Empirical Evaluation of the Simulation Model

In this section we will show some experiments done with no strategies of reorganization which help us to verify some features of the simulation model.

The first aspect we would like to investigate in our simulations is the behavior of crime relative to the so-called impunity factor. In the model, one of the factors for the growth of crime is the increase of the criminals' "quality". This is represented by their experience in committing crimes that increase when they are successful in their initiatives. In other words, if there is no punishment (in this society represented by the prohibition of crime occurrence), the criminals tend to become more dangerous and thus commit more crimes. Simulations without reorganization of routes have shown that such a factor occurs in our model. We run 15 one-month simulations where the police patrol routes are randomly defined and stayed static during the month. We start by examining the following scenario: 15 criminals and 6 police teams and 40 notable

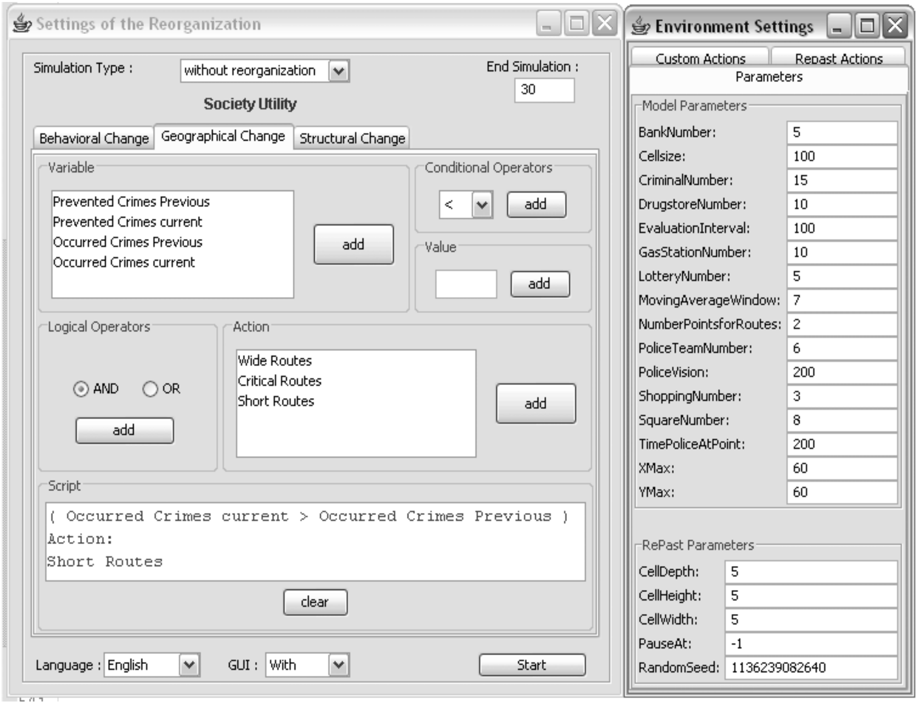


Fig. 1. Interface for configuring the reorganization exemplifies the manner that a logical expression for defining the police commander behavior can be created

points[3]. The probabilities for a criminal to find a certain target are those shown in Table 1 and the other parameters have the values shown in Figure 1.

Figure 2 shows the results for one simulation where no type of reorganization was applied, using these parameters. Several different kinds of interfaces are provided. Graphics show the number of crimes per notable point and the evolution of crimes for such points. The patrol routes are displayed in order to facilitate the accompanying of the reorganizations. In the window where the graphic of Occurred Crimes per Day is displayed, the top line represents the number of crimes that occur per day and the other line represents the moving average of the last 5 days of simulation. For the current example, it is observed that in the course of time, there is a tendency for growth in the number of crimes committed. This occurs due to the fact that with the elapsing of the simulation and with the occurrence of the crimes, the criminals become more experienced and tend to commit more crimes. This result was confirmed after running ten simulations, showing that static routes without reorganization are not enough to control crime. The moving average for all simulations and for different configurations can be seen in Figure 3. Each line represents different proportions of criminals and police teams in that sequence.

[3] This number of police teams and notable points corresponds to the number in a typical neighborhood in the city of Fortaleza from where the criminal data was based on.

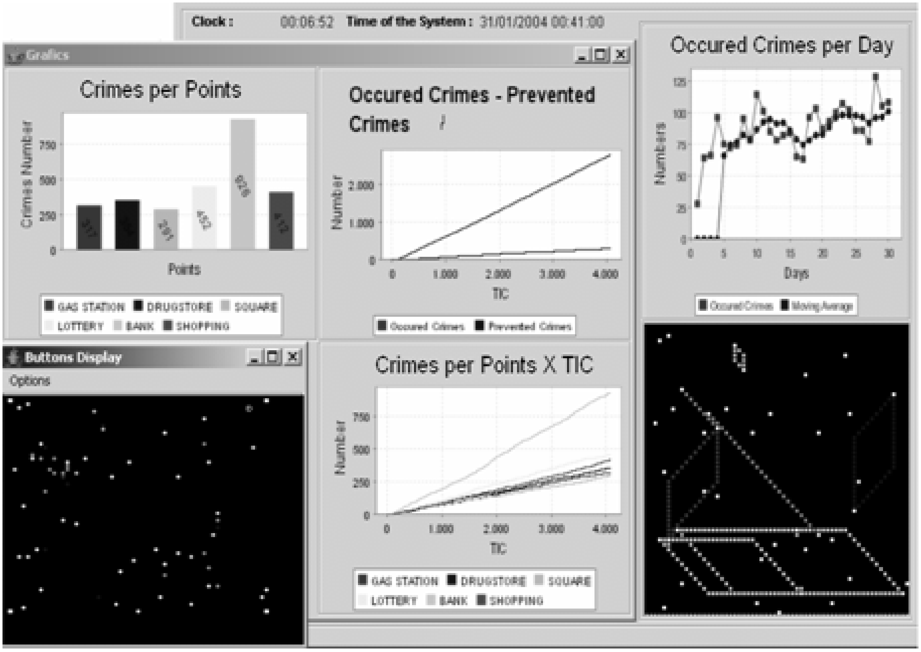


Fig. 2. Overview of the different interfaces. The graphic of crimes that occurred per day in a simulation without reorganization shows the tendency for the crime rate to grow.

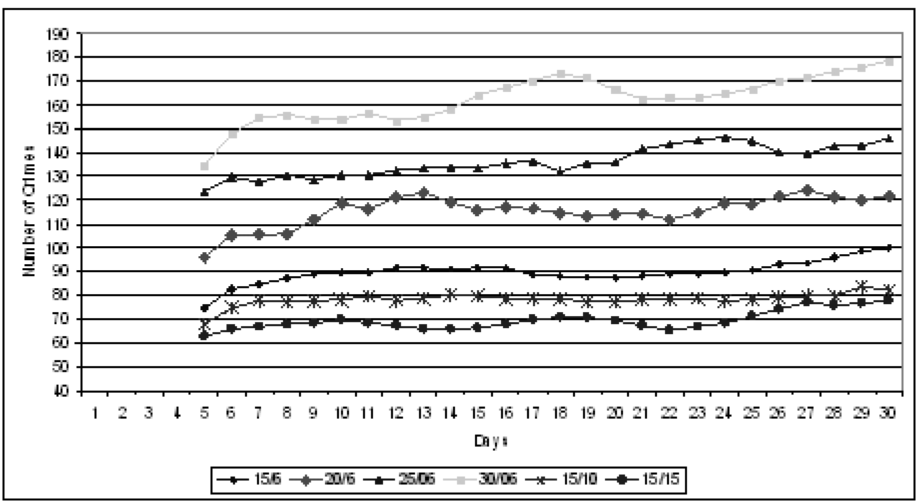


Fig. 3. Different configurations of criminal and police teams with no reorganization

In simulations in which the scenarios had increased the number of criminals, the crime rates are, as expected, higher. Also worth mentioning is that in scenarios where the proportion of criminals to police is less than the scenario initially proposed, even though the crime rates are reduced, the tendency for growth, even though slight, during the month still exists. These results indicate that the increasing of the number of police teams leads to low crime rates but allocating them in a static way is not enough to reduce crime tendency because the criminals tend to adapt their behavior. Such evidence is in accordance with several theories on crime [2] and makes the simulation model stronger.

6 Understanding Physical Reorganization of Agents in the Public Safety Domain

Physical reorganization is used whenever it is necessary to alter the position of the agents without necessarily altering their properties, behaviors, or structural organization. Within the public safety domain, the objective of the society of agents is to minimize the occurrence of crimes. For this, the society falls back upon police teams that organize themselves through routes of preventive patrolling. In this section, we investigate the impact of different strategies of reorganization on the crime rate. Our intent is two-fold. First, we aim at understanding some predefined strategies of reorganization for elaborating future variations of the model towards a more autonomous and decentralized process. Furthermore, by investigating these predefined strategies, we can validate some of the features of the actual model.

6.1 Strategies for Reorganization

The simulation tool allows the execution of different dynamic reorganizations in the routes of police teams. For this, three possible versions for the alteration of the routes were made:

- Short routes are those where notable points are selected and the police teams are designated to these points so that the route of these teams may be the shortest possible. The commander chooses the most vulnerable point and creates a route that leads to this point and n others closest to it. The process is repeated until all the police teams are allocated. With this measure, the notable point will have more time with a police team close to it.
- Wide routes have police teams designated to visit notable points so that the route of these teams may be the longest possible. The commander chooses, from the list of points to be patrolled, the most vulnerable point and creates a route that leads to this point and the n others farthest from such a point. The process is done until all the police teams are allocated. The idea of this strategy is to make the police team cover a larger physical space.
- Critical routes (high criminal activity) have police teams designated to cover notable points identified as the most critical (high crime rate). For each police team to be allocated, the commander chooses the n most vulnerable points and creates a route that lead to them.

The number n of points to be patrolled is a parameter of the simulation, but the new routes to be generated should patrol at least two notable points. In each reorganization, all the police teams are available to be reallocated by the commander. The notable point keeps two related factors "in mind": the number of crimes occurred and the number of crimes that were prevented at this point. These properties are useful to characterize the level of vulnerability of the notable point. The notable points that are candidates for having a police team patrolling its surroundings are those where a greater number of crimes have occurred than have been prevented.

6.2 Applying Reorganization

In this section we exemplify how the reorganization tool is used to aid the understanding of the quality of police routes for the reduction of crime. We start by examining the same scenario that was used when no reorganization was evaluated, namely 15 criminals and 6 police teams and 40 notable points. The probabilities for a criminal to find a certain target are those shown in Table 1.

Figure 4 displays only one graphic of crimes that occurred for a simulation where centralized reorganization was applied and based on a short route strategy. The police patrol covers 2 points and an evaluation of the reorganization is taken every seven days. It is observed here that, over the course of time, there is a tendency for decrease in the number of crimes. We may verify, by analyzing the moving mean line of the simulation, that with the application of reorganization (around days 10, 16 and 23), there was a decrease in the number of crimes and in the growth tendency. This is due to the greater presence of policing at the notable points where crime occurred more often.

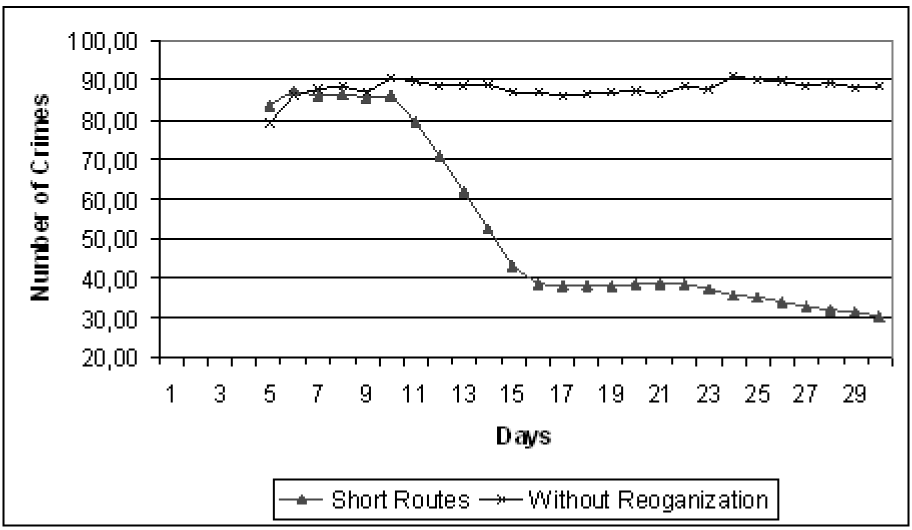


Fig. 4. Crimes occurred with reorganization following the strategy of short routes

6.3 Comparing Patrol Reorganization Strategies

To evaluate the three predefined types of routes, we decided to keep constant the values of several variables so that some comparative results may be obtained. Initially, the only variation was for the type of route. For each type of route, we ran 10 simulations. We started by testing the simulator without any type of reorganization, so that we could better identify the performance of the reorganizations. Later we carried out the same tests with the other three proposed forms of reorganization (short routes, long routes, and critical routes).

The best result obtained for a one-month simulation was a 2-point short route with reorganization evaluation on each day. Figure 5 shows these results. It is possible to observe that all the options with reorganization lead to better performance in terms of crime rate decrease. Crime is under control in all of them and the total number of crimes is also reduced. Figure 5 also shows the crime average total number over the one-month simulation period. The short route strategy leads to the fewest number of crimes in addition to leading the tendency of crime rates to decrease during the month.

We vary the parameters to evaluate routes with different patrol points and a different threshold of reorganization evaluation. The latter parameter defines the moment the commander decides to evaluate whether a reorganization can be done or not. We

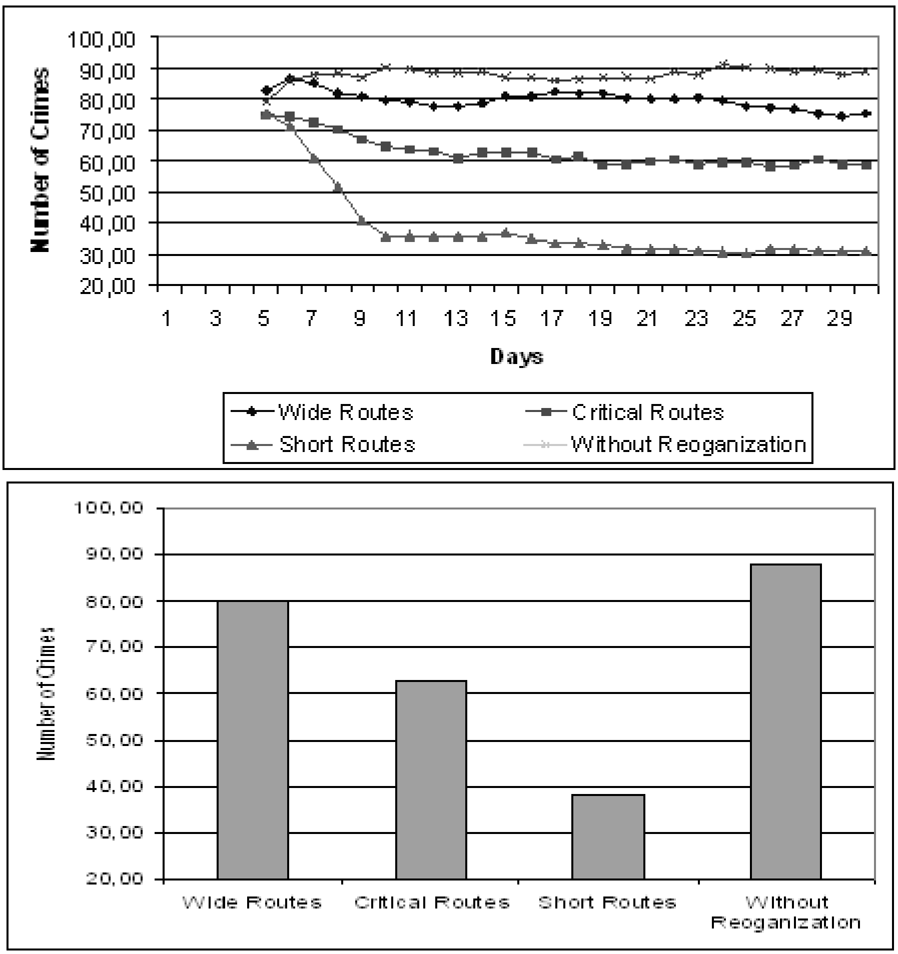


Fig. 5. Comparison of routes

evaluate routes with 2, 4, 6, and 7 points. As for the reorganization evaluation, we ran simulations with 1, 3, 5, and 7 days of delay, i.e. for each 3 days[4] of simulation, the commander analyses the trigger for reorganization and decides whether it should be done or not. The results of these simulations are shown in Figure 6. Moreover, we evaluated the size of the routes for each different time span of evaluation. The results of this comparison are depicted in Figure 7. Figures 6 and 7 refer to the same data but show a different perspective of them. In the x-axis represents the number of points per route and the delay (in ticks) between evaluations of reorganization (i.e. *2-100* corresponds to two-point routes that evaluation of reorganization is done each and every 100 ticks).

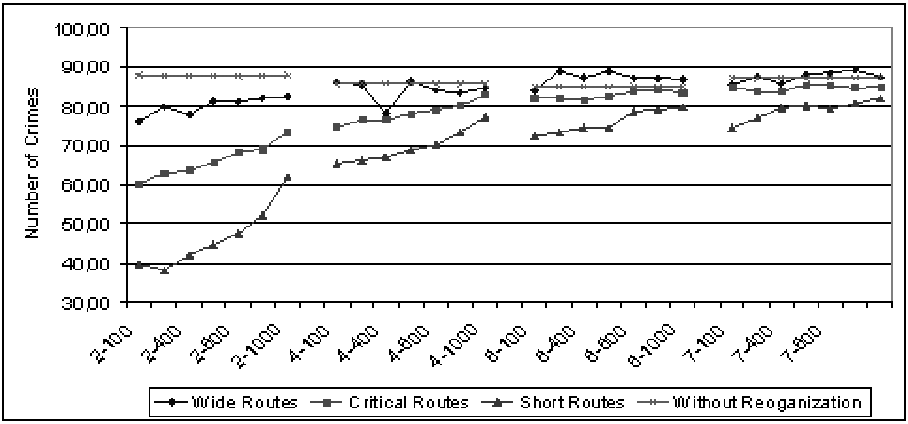


Fig. 6. Comparison of routes for different route size varying the time span for evaluating whether reorganization must be done or not

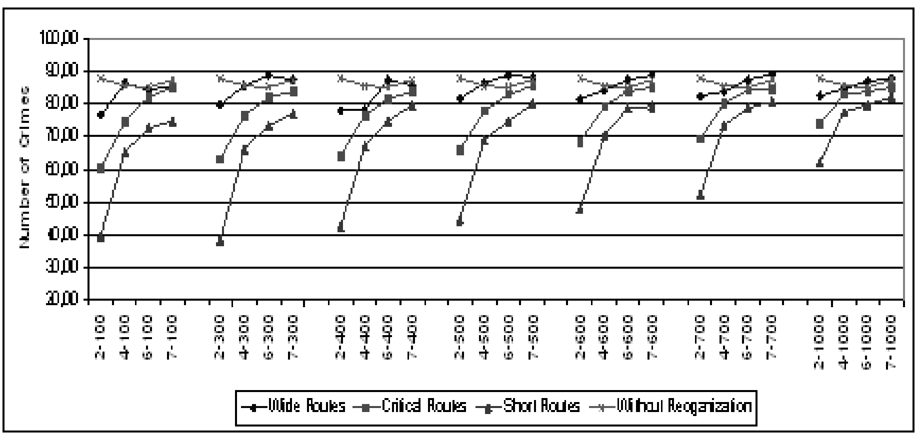


Fig. 7. Comparison of routes for each time span of evaluation varying the police patrol size

[4] One day corresponds to approximately 300 ticks.

In Figure 6, we organize the data by fixing the number of points per route and varying the interval of reorganization evaluation. In Figure 7, we fix the interval of evaluation and vary the number of points per route. These figures make us to understand that the longer the time to evaluate reorganization the greater the number of crimes is, undependably of the route size.

6.4 Discussion

Basically the results of our tests suggest relevant, although preliminary, findings. Regardless of the type of routes, reorganization reduces the crime rates. The fewer the routes, the better the results will be in terms of crime rates as well as in terms of tendency changing. This occurs because we have focused on crimes against properties with fixed targets. So this requires a police strategy that must visit the targets as much as possible. In that case, the time of displacement is very low productive. Experts in policing disagree about the ideal size of police routes. However some heuristics are shared by most of them. Police teams cannot be totally static. They must move in order to be visible and then to bring a feeling of safety. However, when routes are too large and need to be performed with vehicles, this sensation of safety is ephemeral because police pass by but do not stay.

As for the decision to reevaluate it must be as frequent as possible, suggesting that a police team must have a high level of mobility. To reorganize in a time span less than a day was shown unproductive sometimes, even though we did not consider it any cost associated with reorganization. This occurred because the criterion of triggering the reorganization was the increasing of the number of crimes which is a very sensitive one and, if done too often, can provoke unnecessary reorganizations. In real life, reorganization involves a cost of management and displacement, so reorganizing more than once a day is unfeasible. We intend to consider the cost notion in future works.

7 Conclusion and Future Work

In this article, we describe a tool to aid the configuration of agent reorganization strategies. This tool allows for the implementation of behavioral, structural, and physical reorganization strategies. In particular, the article concentrates on physical reorganizations and exemplifies its use in the public safety domain. A society of artificial agents in this domain was modeled, inhabited by criminals and police teams. The health of the society is measured by the number of crimes that occur. Police patrol routes are the main variables for crime to be kept under control. The tool supplies functionalities for the configuration of different reorganization strategies, allowing an analysis of the relationship between the environments that compose the society and the policing strategies. Examples of how reorganizations can influence crime rates are given. The ideas made explicit in the article are the fruits of an ongoing work, where new fronts are being explored. The first front refers to automatic learning of when and how to restructure. In the current version of the tool, these decisions are strongly influenced by information supplied by the user and/or by the designer. Genetic algorithms are being investigated with the intention of supplying more autonomy to the society of agents. By resorting to evolutionary computation resources, our main objective is to automatically uncover effective police patrol routes for coping with certain preconceived scenarios of crime occurrences that typically arise in large urban centers. That is, the idea is not to design such routes by hand, but to let them emerge as a direct result of the application of a customized genetic algorithm approach.

The second investigation front refers to the application of the tool in deeper analyses of criminal issues. A study on the several factors foreseen in the society of agents and how these factors interrelate with the patrolling strategies is being carried out. Specialists in the area of public safety are participating in the project and are being consulted on the generated results. We are investigating how the swarm intelligence concept can be useful to model the experience of a criminal and the attractiveness of a target. We concentrate our studies on strategies that demonstrate some level of self-organization on the modeling of criminal activities. The idea is to model the criminals to make stochastic decisions about the points to attack based on the point's proximity as well as their level of experience about the specific points. Finally, we are studying the impact of the cost on the reorganization process. In public safety, reorganizations cannot be carried out at any time or too frequently.

Artificial society, in the way it is represented, also allows for the investigation of the effect of impunity, although this is not directly represented in the society of agents. Actually, the growth of criminal activity comes from the increase of the criminals' "quality". This is represented by their experience in committing crimes whereby the "quality" increases when they are successful in their initiatives. In other words, if there is no punishment (in this society represented by the prevention of occurrence of crime) the criminals tend to become more dangerous and thus commit more crimes.

References

1. Almeida, A., Ramalho, G. L., Santana, H. P., Tedesco, P., Menezes, T. R., Corruble, V., Chevaleyre, Y. Recent Advances on Multi-Agent Patrolling. In Advances in Artificial Intelligence – SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil. LNAI 3171, Springer-Verlag, 2004.
2. Becker , G.: Crime and Punishment: An Economic Approach. The Journal of Political Economy 76: 169-217, 1968.
3. Benenson, I. and Torrens, P.M. Geosimulation: object-based modeling of urban phenomena. Computers, Environment and Urban Systems. Computers, Environment and Urban Systems 28 (1/2): 1-8, 2004.
4. Carley, K. and Gasser, L.: Computational organization theory. In G. Weiss, editor, Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, pages 299–330. The MIT Press, 1999.
5. Cohen, L., Felson, M. (1979),"Social change and crime rate trends: a routine approach". *American Sociological Review*, 44: 588-608
6. Collier, N. Repast: An extensible framework for agent simulation. In http: repast.sourceforge.net, 2003.
7. Dignum, V., Dignum, F., Sonenberg, L. Towards dynamic organization of agent societies. In G. Vouros, editor, *Workshop on Coordination in Emergent Agent Societies, ECAI 2004*, pages 70–78, 2004.

8. Dignum V., Dignum F., Furtado V., Melo A., Sonenberg L.: Towards a Simulation Tool for Evaluating Dynamic Reorganization of Agents Societies. Workshop on Socially Inspired Computing. AISB Convention 2005.
9. Hannebauer, M.: Autonomous Dynamic Reconfiguration in MultiAgent Systems, volume 2427 of *LNAI*, SpringerVerlag, 2002.
10. Furtado, V., Vasconcelos, E.: A Multi-Agent Simulator for Teaching Police Allocation. Proceedings of the 17^{th} Innovative Applications of Artificial Intelligence, IAAI-2005, Pittsburgh, USA. July, 2005.
11. Garcia, A. C. B., and Sichman, J. S. *Agentes e sistemas multiagentes.* In Sistemas Inteligentes: Fundamentos e Aplicaçoes, S. O. Rezende, Ed. Editora Manole Ltda., Barueri, Sao Paulo, Brazil, 2003.
12. Gutin, G, Punnen, A.P., The Traveling Salesman Problem and Its Variations, Series Combinatorial Optimization, vol. 12, Springer-Verlag, 2002.
13. Valetto, G., Kaiser, G. and Gaurav S. Kc, ‘A mobile agent approach to process based dynamic adaptation of complex software systems’, in 8th European Workshop on Software Process Technology, pp. 102–116, (2001).
14. Winoto, P. A Simulation of the Market of the Offenses in Multiagent Systems.: Is zero Crimes Attainable ? Multi-Agent-Based Simulation II, Bologna, Italy, 2002.

Agent-Based Geo-simulation to Support Human Planning and Spatial Cognition

Nabil Sahli and Bernard Moulin

Laval University, Computer Science Department,
Ste-Foy, G1K7P4 QC Canada
{nabil.sahli, moulin}@ift.ulaval.ca

Abstract. In this paper we emphasize the strengths and weaknesses of human planning and especially within a geographic space. We propose a multi-agent simulation approach in order to overcome some of these limitations while reasoning about a large-scale geographic space. A cognitive complementarity between software agents and human beings emerges from this approach. An illustration on wildfire fighting is presented.

1 Introduction

Human planning requires the simulation of plans. More precisely, Hoc claims that human planning is based on anticipation and schematisation [14]. These aspects have been studied by psychologists for a long time and formalized by the AI community.

When a person faces a new situation, she is able to detect similarities with well-known situations and to anticipate events or properties which are not yet totally identified [14]. Thanks to her sense of anticipation, a person is able to take into account future events while making a decision. In most cases, she anticipates future courses of action in a schematic way by using pre-acquired knowledge. Anticipation is strongly linked to the schematisation process especially for complex resolutions. The notion of schema was introduced by Bartlett [4] as an active organisation of past reactions or anterior experiences. Schematisation is an inference tool of anticipation from previous cases.

According to Craik [6], planning is a very refined activity implying a mental formulation of future states of the world. A human person manipulates her mental representations in order to simulate the real world behaviour and then to make predictions in order to avoid both errors and possible risks.

Nevertheless, human planning capabilities are limited. In [18], Kahneman and Tversky claim that simulation is difficult for humans, which is proven by Forbus' experiences [12] which show that simulation cannot be done by humans, expect in trivial cases.

Planning becomes more complex when addressing uncertain situations. In such a case, accurate predictions about plan executions remain a hard task for human planners. The AI community proposed solutions such as the *Simulation-Based Planning* approach (SBP) [21] which consists of associating planning and simulation. Each generated plan is simulated in order to be tested and evaluated. The most appropriate plan is kept.

J.S. Sichman and L. Antunes (Eds.): MABS 2005, LNAI 3891, pp. 115 – 132, 2006.

The problem is even more complex when considering spatial constraints. In this paper, we focus on spaces as defined in the *Naïve Geography* theory [10]. The geographic space is a large-scale space which goes beyond the human body and which can be represented by different geometries and scales. Therefore, people explore the geographic space by navigating in it and by modeling it according to various points of view which are gathered (mentally) as a puzzle. This makes the geographic space different from a small-scale space in which objects can be handled and the viewer can move, touch and measure these objects in order to collect relevant information.

Forest fire fighting is a typical problem involving planning within an uncertain real world under strong spatial constraints. In this project, we investigate how SBP can help solve this type of planning problem based on a multiagent geo-simulation approach [27].

In Section 2, we discuss the limitations of both human and agent spatial cognitions. We introduce the wildfire domain in Section 3. In Section 4, we propose a geo-simulation architecture based on software agents, which aims to solve complex planning problems in large spaces such as the case of forest fighting. In Section 5 we present the implemented system. We discuss our result in Section 6.

2 Human Spatial Planning and Agent Spatial Planning

In order to plan within a large-scale space, a person relies on her mental representation of the space which is complex [6]. There is good evidence that the capability of planning and moving in a spatial environment is based on the use of *cognitive maps* which are mental constructs that we use to understand and know environments and use to make spatial decisions [19]. This is definitely a sophisticated mechanism.

Geographic reasoning is typically based on incomplete information. However, human beings are once again able to draw quite accurate conclusions by cleverly completing the information or by applying certain default rules based on common sense. Cognitive studies such as [26] reveal that humans use schemas which are hierarchically organized in order to reason within a geographic space and compensate for the lack of information.

Despite this sophistication, *spatial cognition*[1] remains very limited primarily at the level of perception. According to Lynch [25], topology is the first class of information within a geographic space whereas metrical properties, such as distances and forms, are used for refinement purposes and are generally captured in a less accurate way by human beings. Lynch also demonstrated that errors related to human cognitive maps are usually metrical and hardly topological. Here are some sources of metrical errors:

- *Distortions*: when a person forms her cognitive map, the brain uses some heuristics to easily create and record the cognitive map [37]. This causes distortions since the use of heuristics can often reduce the accuracy of the stored cognitive map. The experiments of Stevens and Coupe [36] demonstrated that information within cognitive maps do not form a continual map but rather hierarchical structures (like an atlas). The city of Reno (in Nevada) is then wrongly thought to be to the East of San Diego (California) because Nevada is to the East of California.

[1] An internal reflection and rebuilding of the space in our mind [19].

Lloyd and Helvly [23] suggested three factors which may cause distortions: topographical factors, human factors (their type of activity, their familiarity with the environment, etc.), and treatment factors (heuristics used in order to simplify and to structure spatial data).

- *Distances*: Rothkegel and his colleagues [33] distinguished two types of distances: egocentric distances (distances having the observer as origin) and exocentric distances (distances between two objects different from the observer). For the egocentric distances, researches show that the visual space is not a linear transformation of the physical space [24] and that short distances are often overestimated whereas long distances are underestimated [38]. With respect to exocentric distances, Rothkegel and his colleagues [33] claimed that distances corresponding to a line within an oblique plan are more difficult to estimate than distances in vertical or horizontal plans.
- *Directions*: People often have difficulties to determine geographic orientations (North, East, West and South).
- *Naïve Geographic space is two-dimensional and the reality is three-dimensional*: All objects, even small and thin, have three dimensions. In the opposite, space for humans is two-dimensional [10]. For instance, people often overestimate the steepness of slopes, and the depths of canyons compared to their widths. So, instead of parsing a three-dimensional space into three independent one-dimensional axes, geographic space seems to be interpreted as a horizontal, two-dimensional space. The third dimension is reduced to an attribute rather than an equal dimension.

To conclude, a human planner, despite his exceptional cognitive capabilities, has several limitations when trying to plan in a real large-scale geographic space. We wondered if an agent geo-simulation approach would help overcome some of these limitations.

Agent technology proved its usefulness in several domains such as e-business and planning problems. Besides, the agent paradigm seems to be quite beneficial for spatial reasoning. In [20], Kray enumerates several advantages of using multi-agent systems in a such reasoning, but several obstacles need to be overcome:

a) Planning using agents imposes instead of proposing: A plan proposed by an agent, even if based on real constraints and formal calculations, is often rejected by a human decision maker because it does not comply with common procedures, seems infeasible (according to experience) or simply because it does not fit with the person's expectations.

b) Incomplete use of the agent paradigm in spatial reasoning: In the spatial reasoning literature, the term "agent" is often used to describe systems which perform specific tasks such as planning an itinerary (for example, [32]). However, these agent-based systems usually cannot be considered as true multi-agent systems MAS [20]. In fact, MAS are designed and implemented as systems composed of several interacting agents [16]. Unfortunately, we cannot yet see all the advantages of true MAS in the spatial reasoning since most of the proposed systems do not take advantage of the interactions between agents [20]. More recently, Drogoul and his colleagues [9] claimed that according to their studies, agents' properties (autonomy, proactivity, and interaction) defined at a metaphorical level, are not translated into computational properties.

c) Weak geo-spatial awareness: We have already emphasized this issue in [28]. Nevertheless, most of current research works such as [2] do not propose practical and efficient solutions to provide agents with sufficient geo-spatial awareness. This is mainly due to three factors:

- *Spatial awareness mechanisms are usually "too" simple*: several works which evoke spatial awareness consider "space" as a small spatial representation instead of using the richness of real geographic data. Such geographic data can be manipulated using Geographic Information Systems (GIS).
- *Space scale*: Most current works address small-scale spaces. Problems are certainly different in large-scale spaces.
- *Lack of realism*: MASs which plan (or reason) in a geographic space, lack realism in most cases. The geographic space which is very complex by nature, is often reduced to a two-dimensional space with two or three characteristic properties. Spatial details are often neglected, which leads to a lack of realism.

Agarwal and Abrahat [1] are one of the rare research groups who have been interested in promoting this aspect in the agent community. They proposed to incorporate new capabilities of reasoning into agents as well as a cognitive framework for geometry, topology and spatio-temporal relations.

d) Cognition: Software agents in current systems seem to have difficulties with spatial planning. They do not possess refined cognitive capabilities as humans have. This may probably explain why many planning systems, mainly used to manage crisis situations, were abandoned by users and replaced by classical manual methods using paper maps.

In Section 4, we propose a new approach of simulation-based planning which uses software agents with better spatial awareness mechanisms. This approach also brings agents' capabilities and human cognitive capabilities (especially anticipation and spatial reasoning) closer and use accurate spatial data to achieve an acceptable level of realism.

3 Forest Fire Fighting

As an illustration of our approach, we look on a decision support system for wildland fire suppression actions. Removing the fuel source is the most common method of attacking wildfires. This method does not extinguish the fire. The fire continues to burn until the available fuel is consumed. Removal of fuel in the path of the fire prevents the fire from spreading past the fireline. Depending upon the fire size, the attack may be either direct or indirect. On smaller fires, firefighters usually attack from the flanks and extend their perimeter toward the head until the fire is surrounded: this is the *direct attack.* The lines should be located at the edge of the fire or parallel to it with plenty of escape routes [11].

When the fire is more intense, an *indirect attack* is used. Firefighters (using dozers) clear a line (firebreak) down to the bare soil along a course that runs roughly parallel to the perimeter of the fire but at a safe distance from the fire line. Controlling the perimeter requires anticipating the fire's behaviour and making skilful use of natural barriers (incorporating them as part of the firebreak) [11]. For example, building the firebreak usually starts from an anchor point and goes through light-fuelled terrain and lakes as much as possible. Sharp angles must be avoided when building the line since they can cause spotting (when a fire crosses over the line) as the wind may blow the fire over the line. In practice, fire managers are responsible for choosing where to roughly build (by drawing a sketch) the firebreak and how many resources (dozers) should be deployed. They may then try to use as many anchors and light-fuelled spaces as possible.

We aim to assist fire managers using a virtual environment – virtual, but as realistic as possible – in which software agents can find a suitable path for dozers. A suitable path means that it follows a manager-made sketch as closely as possible, that it takes into account the fire propagation over time, and that it tries to pass through easier terrain while respecting particular constraints (e.g. No sharp curves).

4 EKEMAS: A Multi-agent Approach for Simulation-Based Planning

To make it more understandable, we illustrate our approach on the wildfire fighting case. The approach consists in drawing a parallel between the Real Environment (RE) (i.e. a forest in fire) and the Simulated Environment (SE) (i.e. a virtual reproduction of the forest in fire). For an acceptable level of realism, spatial data within the SE should absolutely come from a GIS since we are working on geographic large-scale spaces. Actors such as firefighters lack of information and capabilities when planning in the RE. Human limitations on both spatial cognition and simulation prevent firefighters from respectively apprehending the space (the forest and the fire) in an accurate manner and from planning complex actions which take into account the fire spread over time. To deal with this problem, we propose to plan using agent-based geo-simulation. This kind of geo-simulation is a simulation based on GIS data within the SE (which has an enhanced knowledge [27]).

The multi-agent geo-simulation allows a quite realistic simulation of the plan so that human decision makers can visualize the probable consequences of the execution of a plan. They can thus evaluate the plan before it can effectively be executed. For instance, fire managers can simulate the dozers building a firebreak, taking into account the fire spread. If the managers consider the eventual path acceptable, they can send this plan to firefighters who are combating the fire.

Our approach relies on agents which reason about space within the SE. To achieve a sufficient spatial awareness, agents should have advanced capabilities (cf. Section 5).

4.1 EKEMAS Architecture

Based on the ideas presented above, we propose the Framework EKEMAS, Enhanced Knowledge Environment based on MAS which consists of four layers solidly related to the RE.

We briefly present the four layers of our architecture in general as well as its application to our fire fighting case.

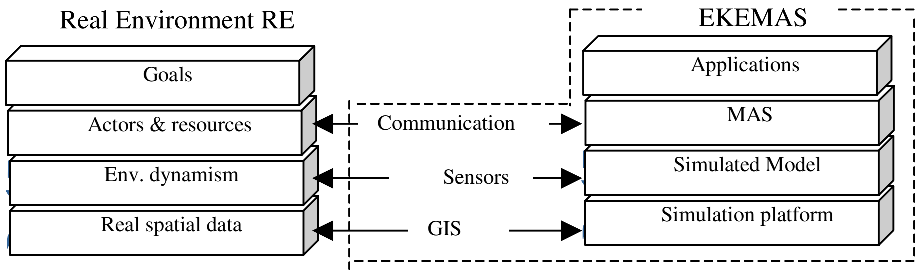


Fig. 1. General architecture of EKEMAS

1st layer: A GIS is essential to reproduce real spatial data in the SE. However, it is not sufficient to create a navigable environment for software agents. Human users also need a visual tool to supervise plan execution. It is thus necessary to transform GIS data into a simulation platform which is visual for human users and navigable for software agents. For the forest fire example we use a platform called MAGS [27] developed by our research group. MAGS allows us to simulate thousands of cognitive agents in moving within a SE in real-time (in 2D and 3D). The spatial environment and the static objects are generated from data contained in GIS and related databases. MAGS agents are equipped with cognitive capacities (navigation, perception, memory, communication, and objective-based behaviour) which allow them to evolve autonomously during the simulation. The SE is divided into a grid of cells. Each cell represents a real square of the terrain (a $25m^2$ square in our implemented sample). It embodies all sorts of relevant information for firefighting: elevation, percentage slope, slope direction, fuel type, etc. Lakes, rivers and roads are also represented since they are important when fighting fires. Lakes and rivers are a source of water and a good natural firebreak; roads are used to transport firefighting resources and sometimes used as firebreaks (depending on their width). Each cell also contains information indicating whether the square is located in a lake, in a river, in a gravel road, etc. All these data come from a GIS.

2nd layer: Nevertheless, a GIS cannot describe all the spatial data that influence the environment. External factors such as natural phenomena make the environment much more dynamic and unpredictable. Models (such as physical models for wildfires) have to be used to simulate this dynamism. Data used by these models should first be captured from the RE and then continuously updated. Sensors may take in charge these two functions. They could be persons (in the terrain), satellites or sensors web [7], etc. The model can thus provide a reliable and progressive simulation of the environment. In our example, we have dynamic data to model. The fire progression is in the form of fire perimeters. We use the Prometheus library [30] to model fire spreading. Prometheus is the Canadian Wildland Fire Growth Model Project which investigates the growth of wildland fires under various conditions. The foundation for the Prometheus model is the Canadian Forest Fire Behaviour Prediction (FBP) System as well as the most recent wave propagation fire algorithms developed at Brandon University. The FBP System is a complex, semi-empirical system that mathematically expresses and integrates many of the fuels, weather, and topographic factors influencing forest fire behaviour. To process Prometheus' data in a program (such as MAGS), we call a COM interface which gives us access to the simulator engine. Prometheus provides at each period of time the expected perimeters of the growing fire. Each perimeter is given as a set of vertices and each vertex embodies the fire's characteristics at this point (the geo-referenced position, the fire intensity, the rate of speed, etc.). We model these vertices using software agents. Each agent represents a fire vertex and embodies its main features.

3rd layer: This is the multi-agent layer. It represents actors (those who perform actions in the real terrain) of the RE. We should have software agents in the terrain (Concretely, these could be electronic devices which sense their neighbourhood and in which agents are embedded) and others in the SE. Agents within the RE may then communicate with agents within the SE, which guarantees a better coherency between data collected from both RE and SE.. Each actor should have a software agent as a representative within the SE. In our example, an actor is a firefighter (or a dozer) who would possess a mobile platform in which an agent is embedded. This agent called *Real Actor Agent* (RAA) interacts with the firefighter via an interface and communicates with its representative (called *Simulation Actor Agent* SAA) in the SE via remote messages or by migration. Before they can act (navigate, perceive, etc.) within the SE, SAA need to be coherently linked to RAA. Fortunately, technology provides many tools that enable communications between agents. In EKEMAS, we only focus on the principle: The RAA has to automatically and periodically notify its correspondent (SAA) about its real position. The SAA moves then accordingly within the SE. For example, when the dozer moves, its new position is sent to the SE via a mobile device (the RAA) which is equipped with a GPS. The SAA can in turn communicate with its correspondent to transmit relevant information from the enhanced knowledge environment to the RE. For instance, a *Simulation Dozer Agent* (a SAA) representing a certain dozer within the SE may send to the real dozer the plan to follow. Interaction modes between RAA and SAA as well as technologies that can be used, are not detailed here.

4th layer: The three previous layers provide a foundation to applications aiming to assist actors within the terrain as well as users of the SE with the goal of making

Table 1. Instanciation of EKEMAS for the Forest Fire case

EKEMAS	Forest Fire Case
Real spatial data	Data about the forest in fire
Environment dynamism	Spreading fire
Actors	Forest firefighters
Resources	Houses, inhabitants, etc.
Goals	Fight the fire and protect resources
RAA	Real Dozer Agent
SAA	Simulation Dozer Agent
Simulated model	Prometheus (simulator for fire spreading)
Simulation Platform	MAGS
Sensors	Satellites, planes, sensor webs, etc.

better strategic decisions. These applications are located in the fourth layer. For our example, the goal of firefighters is to find a suitable path for dozers in both direct and indirect strategies. The corresponding application aims to assist fire managers when planning these paths. Table 1 lists relevant entities of the forest fire case and their corresponding elements within the EKEMAS architecture. We will see in the next section which capabilities agents (of 3rd layer) should possess so that they can support spatial planning (in the fire fighting case).

5 *Pathfinder Agent:* A Spatial Planner Agent

In a real situation while constructing a firebreak, several resources and actors are involved. As mentioned in Fig. 1, we represent each actor and each relevant resource by an agent. For example, a lake (a source of water) is represented by an agent (named *lake agent*) which embodies the lake features (depth, area, etc), a dozer is represented by an agent (named *pathfinder agent*), etc. Fire is also modelled by agents. Each vertex is represented by a software agent (named *fire agent*) which embodies the main characteristics of the fire at this vertex (intensity, rate of spread, direction, etc.) All these software agents (*lake, pathfinder, fire agents*, etc.) interact with each other in order to collaborate or to exchange information. For instance, a *pathfinder agent* perceives *fire agents* located in its neighbourhood and then interacts with them in order to decide how to move safely (far from fire). When building a firebreak, two dozers may be involved. In this case, each *pathfinder agent* should start from an anchor point (the two anchor points will be located at the two ends of the firebreak) and try to reach the other one so that the fire can be surrounded. The two agents must collaborate in order to reach the same final point. Each *pathfinder agent* relies on its capabilities (especially perception) to detect the other *pathfinder agent*.

In this section, we only focus on the implementation of the *pathfinder agent* (in charge of finding a path for dozers) since it is the most important agent in our application. For simplification purposes, we suppose that we have only one dozer within the terrain. The architecture of the proposed solution for the direct and indirect strategies is presented in the Fig. 2.

A *pathfinder agent* can operate in two different modes: ***Pre-execution*** mode and ***Real-time*** mode.

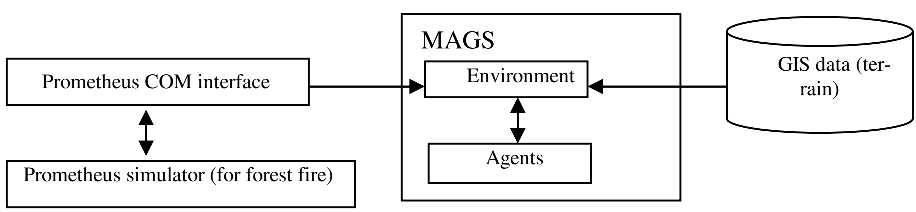


Fig. 2. Our solution architecture (Only the relevant MAGS components are represented)

5.1 Pre-execution Mode

A decision maker may test different strategies by simulating different plans before any effective execution. For example, he may test a direct attack strategy by asking the *pathfinder agent* to follow the fire spread progression as predicted by Prometheus (Fig. 4.a). The resulting path is evaluated by the manager and may or may not be adopted. With respect to the indirect attack, the manager may ask the *pathfinder agent* to look for a path which best fit with the sketch that he has been proposed via a visual tool that allows the user can add a sketch to the simulated environment.

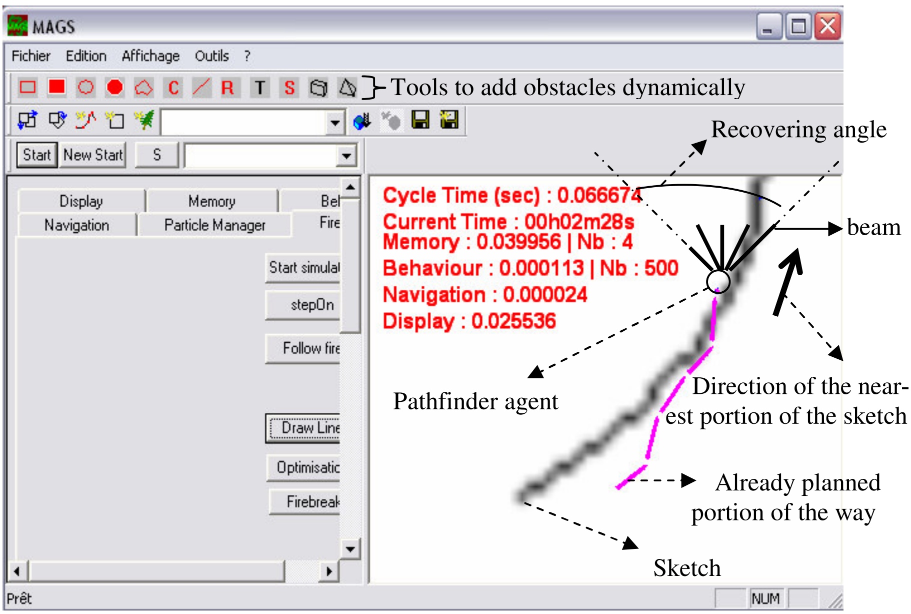


Fig. 3. *Pathfinde agentr*'s procedure to follow a sketch (illustrated in a snapshot)

In our example, the *pathfinder agent* is a MAGS software agent which has to perceive the environment described above, plan its next move, and move in order to reach the planned destination. The purpose of the *pathfinder agent* is to find a suitable path which best follows a sketch made by a fire manager and which respects certain constraints (no sharp curves, etc.). A *pathfinder agent* starting near a sketch end, plans its path as follows:

(1) Detect the direction of the nearest portion of the sketch.
(2) Perceive the environment by launching a beam to a certain distance and according to the direction determined in (1) as well as the nearby directions (the angle between the direction which is the most to left and the one which is the most to the right is called *recovering angle*).
(3) Process the cost of each beam (the cost of moving the dozer according to this beam).

(4) If no beam has a reasonable cost (an obstacle is on its way) then enlarge the recovering angle (that considers more directions) and go to (2) else go to (5).
(5) Choose the less costly beam and move to the destination reached by this beam.
(6) Go to (1).

In step (5), choosing the less costly beam implies taking into account terrain features, spatial constraints and fire nearby. The terrain features which influence the cost of the move are: elevation, slope, slope direction, and fuel type.

Since the objective of the *pathfinder agent* is to follow a sketch without creating any sharp curves, the deviation from the sketch direction and the neighborhood of the *pathfinder agent* to the sketch are the two main factors which influence the beam cost.

Finally, fire closeness, fire spreading direction and fire intensity may present a danger for dozers. The *pathfinder agent* has to take these parameters into account while choosing the best beam to follow. Therefore, a beam which leads a dozer close to the fire line should be more costly than another one which keeps the dozer away from the fire.

If we call *BeamCost* the cost of a beam, *TerrainCost* the cost related to terrain features, *SpatialConstraintsCost* the cost related to spatial constraints, and *FireClosenessCost* the cost related to fire closeness, *BeamCost* is given by the following expression (1) :

$$\begin{aligned}\text{BeamCost} = \ & \text{TerrainCost (elevation, slope, slope\_direction, fuel\_type).} + \\ & \text{SpatialConstraintsCost(deviation, sketch\_nearby)} + \text{FireCloseness-} \\ & \text{Cost(fire\_closeness, Fire\_Spreading\_direction, fire\_intensity)}\end{aligned} \tag{1}$$

The *pathfinder agent* has also to deal with another spatial constraint: the fact that neither a specific starting point nor an accurate ending point is given. Since a starting point can be any point of a certain geographic zone, we have to consider all of them. This would increase our chance to find the best path for the problem. In fact, starting from a certain point rather than another one may spare the *pathfinder agent* a rough

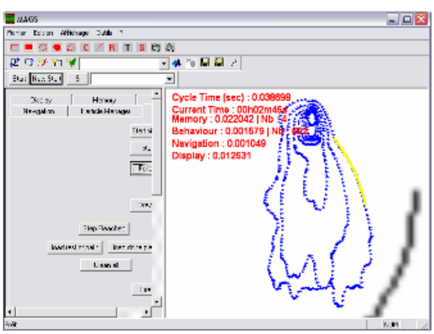


Fig. 4a. The agent simulates a dozer that follows fire perimeters

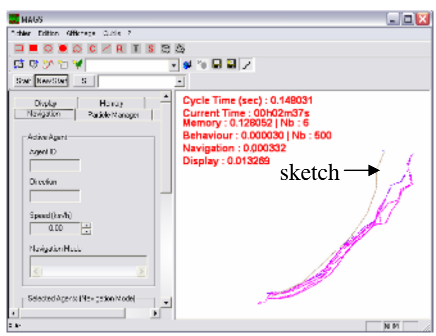


Fig. 4b. A human expert draws a sketch which is the improved by agents

portion of the terrain (for example if the later starting point takes the agent through a large obstacle), which could increase significantly the cost of the path. If *S* is the set[2] of possible starting points, the *pathfinder agent* has to look for a path from each element of *S*. Since we are in a virtual world, we can create a clone for the *pathfinder agent* for each element of *S* (Fig. 4.b). This may ensure looking for the best path in shorter time. Finally, the *pathfinder agent* considers that it has reached its destination if it is in a certain zone (destination zone).

In order to control the combinatory explosion that can occur while trying to investigate as many different solutions as possible, cloned agents have to collaborate in order to avoid redundant or very similar solutions. When a *pathfinder agent* (or a clone) has to investigate, at a given point, several possible destinations, the number of the clones that can be created each time depends on the global number of the clones present within the environment. For example, if this number is low[3], three or four new clones can be created at that point, only one clone would be generated to investigate the second best probable destinations (the first destination is investigated by the *pathfinder agent* which created the clone). Then, when two clones having the same target direction (the same next step destination) meet, the one which has the higher cumulative cost (the cost of the path already done by this agent) self-destructs. The whole mechanism that controls the cloning process is not detailed in this paper (it is the subject of a future publication).

Classical Pathfinding algorithms such as A* cannot deal with such constraints in contrast with our *pathfinder agent* which has quite advanced mechanisms to solve such spatial problems. In [35] we discuss these issues in much more details.

5.2 *Real-Time* Mode

When the plan is in progress, the *pathfinder agent* is used for a different purpose. First, it has to keep track in the SE of the effective moves of the dozer in the RE. Second, it has to periodically replan the rest of the plan starting from the current position of the dozer. This is done in order to anticipate any problem which could occur due to the fact that the environment is so dynamic that data (such as the effective fire spread) may be different from what was expected ten minutes before. The whole approach is detailed in [34]. Finally, the *pathfinder agent* should replan as soon as an unexpected event is reported. For example, the dozer may notify the agent about an unexpected obstacle. The agent should thus find a new path starting from the current dozer's position taking into account the new obstacle.

The new plan is then sent to the dozer. Depending on the time available, a quick or a refined planning may be done (Fig 6). The diagram of Fig 5 illustrates the functioning of a *pathfinder agent* in a *Real-time* mode. Since the replanning process occurs in a SE, the *pathfinder agent* can create as many clones as necessary in order to investigate the different possibilities of avoiding the obstacle.

---

[2] MAGS can support thousands of active software agents in real-time simulation (and accelerated mode). With the current version of MAGS, we can start thousands of *Pathfinder Agents* in parallel, which is largely more than necessary.

[3] This will depend on the platform. In MAGS, a "low" number of clones means few hundreds.

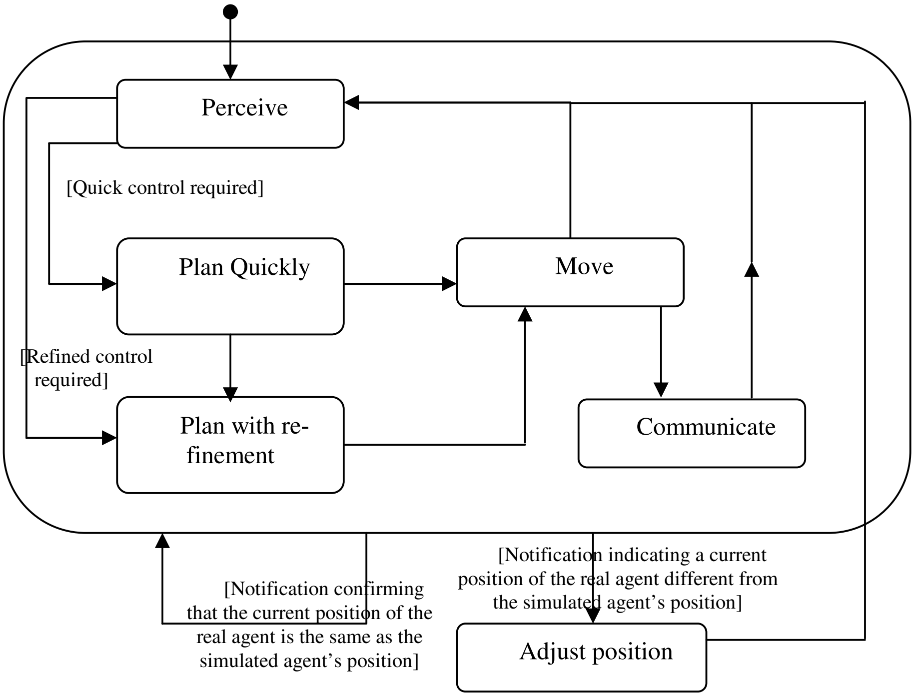


Fig. 5. State diagram describing the functioning of the *pathfinder agent* in a *Real-time* mode

5.3 *Pathfinder Agent's* Capabilities

The *pathfinder agent* relies on several "cognitive" components (Fig. 6) which offer quite advanced spatial capabilities. We detail these elements in what follows.

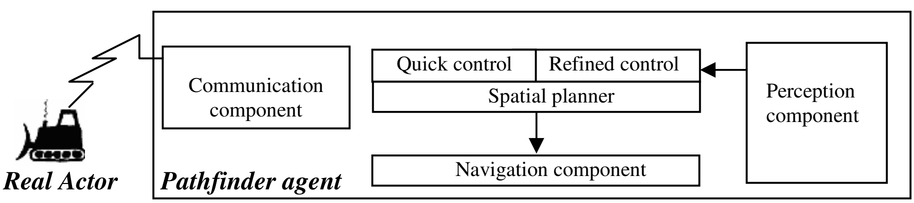


Fig. 6. *Pathfinder Agent* architecture

Perception: Agent perception is usually simulated in SE with the aim to mimic the real behaviour of the simulated entity (e.g. human behaviour). What this kind of agents should perceive has to be as similar as possible to what the real agent would perceive in the RE. In contrast, our *pathfinder agent* uses perception in order to plan. Its perception capabilities should then be stronger than those of real agents since the main objective is not to simulate reality, but rather to take advantage of the SE as an enhanced source of data. It then seems to be more a data access issue than a perception process.

Nevertheless, we will keep the term "perception" since the agent has access only to data from its neighbourhood within a given range of perception.

By analogy to human spatial perception, several perception modes can be assigned to MAGS agents and particularly to *pathfinder agents*: 1) perception of terrain characteristics (elevation and slopes) in the area surrounding the agent; 2) perception of the landscape surrounding the agent (including buildings and static objects); 3) perception of other mobile agents navigating in the agent's range of perception (e.g. a *pathfinder agent* can perceive *fire agents* in its neighborhood and interact with them in order to plan its next moves); 4) perception of messages communicated by other agents.

Spatial planner: After perceiving and before moving, the *pathfinder agent* needs to know, at each moment, what is the best move that it can do. Depending upon the time available and the current constraints, different strategies may be applied.

Quick Control component: Following an unexpected event, this module may assign to the navigation component (described below) a short-term objective. This component is only used in the *real-time* mode. For example, when the *pathfinder agent* has to replan (because of an unexpected obstacle or because the dozer moves differently than planned), it tries to find a quick solution and communicates it to the dozer as soon as possible. This solution is not necessary the best. It may also be incomplete. However, it roughly respects the situation constraints and allows the dozer not to wait for a long time for a complete and better solution. Concretely, when a replanning is necessary, the *pathfinder agent* sends one possible path to the dozer.

Refined Control component: After sending a quick path, the *pathfinder agent* may refine the solution. It can for example try several other possibilities by cloning itself in order to test various strategies. As mentioned before, the cloned agents have to interact in order to eliminate redundant or very similar plans. For example, if an agent detects that another clone has already found a better path leading to the same current position, it destroys itself. The whole mechanism is not described in this paper. If a better solution is found, it is communicated to the dozer. The new solution must of course take into account the current position of the dozer. Unless the fire manager is in a rush, the *refined control* component is also in charge of the planning process under the *pre-execution* mode.

Navigation: This component complements the previous ones. It is in charge of executing the *pathfinder agent*'s moves within the SE as proposed by the *spatial planner*.

Communication: It allows the agent to communicate with the other agents within the SE or the RE. It is also in charge of interacting with the human user.

6 Discussion and Related Works

In the indirect attack strategy, the current procedure consists in deploying firefighters to scout around the space surrounding the dozers. Best paths are then reported to dozers' drivers. The quality of these paths depends upon the perception of the firefighters. However, since these firefighters do not have a global view of the situation (they can just perceive their neighbourhood), they can only propose a piece of a plan instead of a final global plan. This may lead to the plan failure, which is not acceptable in critical situations. Firefighters' decisions, already limited by their human processing capacity, may be distorted as well (due to the limitations of their cognitive map). In contrast, the *pathfinder agent* is able to consider (using its perception capacities) all terrain characteristics in detail in order to find the best path in the SE. Since it performs in a SE (with *enhanced knowledge*), it has a global view of the situation and the terrain. It can thus propose a global plan. However, this agent has to follow the sketch proposed by the manager. To propose such a sketch, the human expert relies on his experience, on his apprehension of the terrain features and on the intuitive reasoning about space (which cannot be performed by agents). We have here an example of how spatial cognitions of both spatially aware agents and human planners can be complementary while using multi-agent geo-simulation.

Since human planning has limitations in terms of simulation capabilities, the *pathfinder agent* offers a good support since it is able to quickly and accurately simulate complex plans. However, given the domain complexity and its nature (real and critical situation), agents need human planners to validate their plans. In fact, software agents do not have a refined sense of anticipation and judgment as a human expert do. Even if the plan seems to be well-grounded, it may not be feasible in reality or may go against certain doctrines, etc. Human experts can propose to change the plan according to their own experience and anticipation sense. This is an example of how agent-based geo-simulation can complement human planning skills when addressing complex problems.

There are several previous intelligent planning applications in the wildfire domain such as PHOENIX [5], CHARADE [3] and SIADEX [2]. Phoenix, for example, is also based on a four-layer architecture quite similar to EKEMAS. The planning process relies on a library of plans. Agents are able to choose the most appropriate plan. Agents are also capable of generating future actions so that they can anticipate failures. Phoenix' agents are organized hierarchically in order to support the hierarchy of actors (firefighters, chief, etc.). In spite of all its features, Phoenix remains a tool for training. It does not propose an extension to support real world situations. Phoenix' agents do not have enough spatial awareness to deal with data about real terrain features and real spatial qualitative constraints. Besides, the planning and replanning mechanism is quite simple and does not take advantage of recent planning approaches such as the *Distributed Continual Planning* [8] which we use in our approach [34]. Even the anticipation of future actions, which is a good initiative, is limited since it does not take advantage of the simulated environment. It would be more judicious if the future actions were simulated instead of only being calculated. In another sort of application (military context) which is quite similar to the wildfire fighting case, Horn and Baxter [15] propose a simulated environment which offers a more elaborated spatial awareness and planning approach (continual planning). Nevertheless, it lacks realism and above all, it does not involve the human planner in the planning process as we do in our approach. Phoenix has the same limitation.

CHARADE, an interactive case-based planning system for wildfires, offers a wide range of tools to assist fire managers when dealing with resources and making plans. However, this system is not based on software agents and does not permit to build plans but rather to find one from a predefined list of already stored plans.

In addition, these works do not provide agents with spatial awareness and do not rely on the complementarity of human and agent planners as proposed in our approach. Our experience with firefighting planning teaches us that such planning problems should be addressed by both humans and agents. In fact, human and software agents have their own limitations, but together they are complementary. We do believe that software agents need more spatial awareness so that they can assist human planners with complex and real spatial problems.

Recently, SIADEX, an intelligent decision support system for the design of forest fire fighting plans proposed to use deliberative planning techniques without the requirement of having a predefined skeleton or library of cases of plans previously stored. It also provides enough interactivity to involve human planners in the planning process. However, this human involvement is not provided at the same level as in our approach in which a human planner can make qualitative suggestions while building the plan. SIADEX does not rely on software agents, but rather on AI techniques, and consequently it does not achieve a sufficient spatial awareness. It does not support real world operations neither (no links between VE and RE) as we do in our approach. Nevertheless, SIADEX deals with several types of uncertainty (temporal and spatial), which is not yet done in our approach.

Finally, Lewis, Lenox, Payne et Sycara [22] have developed a simplified virtual battlefield simulation called MokSAF in order to evaluate how people can interact and obtain assistance from agents within a team environment. Each participant commander has to plan a path with the assistance of software agents called *Route-Planning Agents* (*RPA*). Three types of *RPA* were proposed: *Autonomous RPA* (performs much of the task itself), *Cooperative RPA* (the commander and the agent work jointly to solve the problem), and *Naïve RPA* (provides minimal assistance to the commander). Experiments show that commanders assisted by *Cooperative RPA*, found the best plans. However, participants in the *Autonomous RPA* group expressed frustration with the indirection required to deal with constraints imposed by agents' behaviours. They noticed that they wished they could "just draw the route by hand" [22]. This shows that delegating the whole planning process to agents is not appropriate, whereas complementarity between agents and humans is more beneficial for such complex planning problems. Nevertheless, the level of complementarity proposed by these authors is lower than what we propose in our approach. Actually, we focus more on the spatial and cognitive capabilities of both agents and humans. Even if Lewis and his colleagues' work addresses a different application and different constraints (in [22]: no qualitative constraint is supported, no real data, no parallelism between the virtual and the real worlds, no real-time constraints, agents do not have capabilities such as perception, etc.), it leads to almost the same conclusions as ours. We can summarize them as follows: Software agents are advantageous for the Pathfinding process in real, complex and large-scale environments; Complementarity between software agents and humans is the key for a better agent-assisted planning.

7 Conclusion and Future Work

In this paper, we proposed a new approach of assisting people when solving complex planning problems in real and dynamic large-scale spaces. It is based on a multi-agent geo-simulation approach and draws a parallel between real and simulated worlds. It suggests providing agents with advanced capabilities. In the case of wildfire fighting, we provide agents (*pathfinder*) with geo-spatial awareness to support human planning.

Laboratory experiments of the current version of the software (developed in C++ and applied to wildfire firefighting) show good results. For example, *pathfinder agents* take from few seconds to few minutes (depending on the number of clones used and the path length) to find a path which follows the general shape of a man-made sketch. Such a path cannot be found using classical algorithms such as A*, which provides a highly sinuous path (this is not acceptable when building a firebreak). We have also simulated unexpected events while executing a plan as for example adding a new obstacle or changing the current location of the dozers. The initial plan is generated in few seconds with success. Later, a more refined plan is generated in few seconds or in few minutes (once again, depending on the path length). We also met experts of forest firefighting in Alberta and they were enthusiastic about the project and even convinced that agent-based geo-simulation is the future of fire fighting operations. Nevertheless, using it in real situations will depend on the development of links between the RE and the SE and particularly upon GIS data availability, sensors efficiency, and communication reliability between RE's agents and SE's ones as well as the quality of georeferenced data used in the simulation. Some of these issues are already in a good progress. For example, GIS data are nowadays more and more available and used, some dozers are already equipped with GPS devices to follow an accurate geo-referenced path, etc. These technologies are starting to be applied in similar fields such as in path tracking navigation of autonomous ground vehicle [13]. Siren [17] is a good example of the use of pervasive computing in making real agents (firefighters) profit from the enhanced spatial knowledge that software agents can have.

One of the main issues which are not addressed in our work, is the communication. Nickerson and Lu recently proposed a new language for sensor webs, called SWL (Sensor Web Language) [29]. Software agents may use SWL to communicate with sensor webs. It would be interesting to develop a language as light as SWL and which can support communication between *Simulation Actor Agents* and *Real Actor Agents*. Research in this inter-agent communication has to deal with a number of other challenges such as how to handle quantitative and qualitative spatial information.

Another possible enhancement to our approach would be the integration of a hierarchical structure of agents (especially agents representing actors). Some works, and especially in the military domain such as in [15], have already proposed MASs where agents have different levels in a hierarchy. In the forest fire domain, Phœnix [5] suggested a centralized and hierarchical organization of agents. A *fire chief* agent coordinates the activities of all *dozer* agents by sending directives and receiving reports of the spreading fire. It thus maintains a global view of the situation, whereas *dozer* agents have a local view of the environment based on their own sensorial capabilities.

Finally, and in order to provide each agent with an environment with an appropriate level of detail, the *simulation platform* (the first layer of the EKEMAS architecture) should be multi-scaled. The granularity of the visualized data (in the virtual environment) will depend on the hierarchical level of the agents present within the environment. A possible way to do that is to use a holonical structure for the environment [31].

Acknowledgments

This research is supported by (Natural Sciences and Engineering Research Command) NSERC. The development of MAGS is supported by GEOIDE, the Canadian Network of Centers of Excellence in Geomatics. We gratefully acknowledge the support of Alberta Sustainable Resource Development (ASRD) and SOPFEU (Société de protection des forêts contre le feu) in Quebec province.

References

1. Agarwal, P., Abrahart, R. Agent-based simulation of cognitive neighbourhoods in urban environments, Proceedings of GeoComputation, September 8-10, Southampton, UK. (2003)
2. de la Asunción M., Castillo L., Fdez.-Olivares J., García-Pérez O., González A., Palao F. : SIADEX: an interactive artificial intelligence planner for decision support in forest fire fighting. Workshop on Binding Environmental Sciences and Artificial Intelligence (BESAI 2004), European Conf. on Artificial Intelligence. (2004).
3. Avesani P., Perini A. Ricci F., Interactive case-based planning for forest fire management', Applied Intelligence, 13(1). (2000) 41–57.
4. Bartlett, F.C.: Remembering: A study in experimental and social psychology. Cambridge, England; Cambridge University Press, (1932).
5. Cohen P.R., Greenberg M.L., Hart D.M., Howe A.E.: *Trial by fire: Understanding the design requirements for agents in complex environments*. *AI Magazine*, 10(3). (1989).
6. Craik, K.J.W. *The Nature of Explanation*. London: Cambridge. University Press. (1943)
7. Delin K.A., Jackson S. P., Some R.R., Sensor webs, NASA tech Briefs, 23. (1999).
8. DesJardins M. E., Durfee E.H., Ortiz C.L., Wolverton M.J.: A Survey of Research in Distributed Continual Planning. *AI Magazine*, 1(4). (1999) 13-22.
9. Drogoul A., Vanbergue D., Meurisse T., Multi-Agent Based Simulation: Where are the Agents ? Proceedings of MABS'02 (Multi-Agent Based Simulation, Bologna, Italy, July 2002), LNCS, Springer-Verlag. (2002)
10. Egenhofer M., Mark D.: Naive geography, in proceedings of the international conference COSIT'95, Springer-verlag, Lecture Notes in Computer Science LNCS 988. (1995).
11. Firewise Website: http://www.firewise.org/pubs/ wildfire_control/ updated 2004.
12. Forbus K.D.: A Study of Qualitative and Geometric Knowledge in Reasoning about Motion, MIT Artificial Intelligence Laboratory, Research Report AI-TR-615. (1981).
13. Hellstrom T.: Autonomous navigation for Forest Machines, A project Pre-study in the Department of Computer Science, Umea University, Sweden. (2002).
14. Hoc J. M.: *Cognitive Psychology of Planning*. Academic press. (1987).
15. Horn G., Baxter J.: An Interactive Planner for Tank Squadron Assaults, on the first Planning and Scheduling Special Interest Group *workshop,* Milton Keynes, UK. (2000).
16. Jennings N. R., Sycara K., Wooldridge M.; A roadmap of agent research and development. *Journal of Autonomous Agents and Multi-Agent Systems AAMAS* 1. (1998) 275–306.
17. Jiang X., Chen N.Y., Hong J. I., Wang K., Takayama L., Landay J.A.: Siren: Context-aware Computing for Firefighting, in proceedings of *Pervasive 2004: The 2nd Int.Conf. on Pervasive Computing, 19-23 Linz /Vienna, Austria*. April. (2004)
18. Kahneman D**.**, Tversky A.: The simulation heuristic. In Kahneman, Slovic & Tversky (Eds.) *Judgement under uncertainty: Heuristics and biases*. New York: Cambridge University Press. (1982).

19. Kitchin R. M.: Cognitive Maps: What Are They and Why Study Them?, *Journal of Environmental Psychology*. (1994) 14:119.
20. Kray C.: The benefits of multi-agent systems in spatial reasoning. Special Track on Spatio-temporal Reasoning. Proceedings of the conference *Flairs'01*, Key West, USA, (2001).
21. Lee J. J., Fishwick P. A.: Real-Time Simulation-Based Planning for Computer Generated Force Simulation, *Simulation*, 63:5 (1994) 299-315.
22. Lewis M., Lenox T., Payne T., Sycara K. Spatial Planning in Teams of Human and Machine Agents. In Second National Conference Information Technology and Spatial Planning. Isole Tremiti, It. (2001).
23. Lloyd R., Heivly C., Systematic distortions in urban cognitive maps. Annals of the Association of American Geographers, 77 (1987) 191-207.
24. Luneburg R. K.: *Mathematical Analysis of Binocular Vision*. Princeton: Princeton University. (1947)
25. Lynch K.: *The Image of a City*. Cambridge, MA: MIT Press. (1960).
26. McNamara T. Hardy J., Hirtle S.: Subjective Hierarchies in Spatial Memory, Journal of *Environmental Psychology: Learning, Memory, and Cognition* 15(2) (1989) 211-227.
27. Moulin B., Chaker W., Perron J., Pelletier P., Hogan J. & Gbei E.: MAGS Project: Multi-Agent GeoSimulation and Crowd Simulation. *Spatial Information Theory*, Kuhn, Worboys & Timpf (edts), Lecture Notes in Computer Sciences, vol. 2825 (2003) 151-168.
28. Moulin B. and Sahli N.: Software agents and Distributed Continual Planning: towards geo-spatial awareness, Proceedings of Autonomous Agents & Multi-Agents Systems (AAMAS02), Workshop Agentcities: Challenges in Open Agent Environment, Bologna (Italy) (2002) 164-167.
29. Nickerson B. G., Lu J., A language for wireless Sensor Webs. 2nd Annual Conference on Communication Networks and Services Research CNSR, May 2004, Fredericton, N.B., Canada. IEEE Computer Society (2004) 293-300.
30. Prometheus Website: http://www.firegrowthmodel.com. Last update: sept (2004).
31. Rodriguez S., Hilaire V. Koukam A., Holonic Modelling of Environments for Situated Multi-Agent Systems. The Second International Workshop on Environments for Multiagent Systems. In AAMAS'05, Utrecht, Holland (2005).
32. Rogers S., Fiechter C. N., Langeley P.: An adaptive interactive agent for route advice. In Proceedings Of the international Conference *Autonomous Agents '99*, New York, New York: ACM Press. (1999) 198–205..
33. Rothkegel R., Wender K. F., Schumacher S., Judging spatial relations from memory. In C. Freksa, C. Habel, & K. F. Wender (Eds.), *Spatial Cognition - An interdisciplinary approach to representation and processing of spatial knowledge* Berlin: Springer (1998) 79-105.
34. Sahli N., Moulin B.: La planification agent dans un environnement dynamique et distribuée: entre la réalité et la simulation. Déploiement des systèmes multi-agents vers un passage à l'échelle., Revue des sciences et technologies de l'information. Édition Lavoisier, Hammamet, Tunisie, novembre, (2003) 116-127.
35. Sahli N., Moulin B.: Real-World Pathfinding using Agent-Based Simulation. Agent-Directed Simulation Symposium, part of the SpringSim'05. San Diego, CA. (to appear) (2005).
36. Stevens A., Coupe P. Distortions in Judged Spatial Relations. *Cognitive Psychology* 10: (1978) 422-437.
37. Tversky B.: Distortions in memory for maps. *Cognitive Psychology*, 13, (1981) 407-433.
38. Wender K. F., Wagener-Wender M. Rothkegel R.: Measures of spatial memory and routes of learning. *Psychological Research*, *59*, (1997) 269-278.

Modeling Non-linear Common-Pool Resource Experiments with Boundedly Rational Agents

Eva Ebenhöh

Institute of Environmental Systems Research, University of Osnabrück, Germany
Eva.Ebenhoeh@usf.uos.de,
http://www.usf.uos.de/∼eebenhoe

Abstract. This paper presents a model of non-linear common-pool resource experiments with boundedly rational agents. The model is based on data of experiments with and without communication and reproduces individual and not only aggregated data. It is part of a framework for modeling economic experiments. The agents exhibit bounded rationality in the sense that they use simple heuristics and simple learning processes in an aspiration adaptation process, and base their decisions on norms and "'emotions". A major objective of this kind of modeling is to implement agents in a way that makes it possible for humans to identify with agent behaviour.

1 Introduction

Understanding human behaviour in social dilemmas is challenging. A social dilemma exists, when individual rationality differs from group rationality. This happens, for instance, during the appropriation of common-pool resources, like fishing in a community lake. Common-pool resources are characterised by subtractability and difficult exclusion; in the example of a lake that means fish extracted by one fisher cannot be caught by another and it is difficult to keep community members from fishing. In this example individual and group rationality would lead to different behaviour, if it is individually profitable to fish more than the resource can sustainably provide. Early work on social dilemmas induced by the joined use of common-pool resources include Demsetz [1] and Hardin [2] who propose to solve the dilemma of these situations by changing porperty rights. If an "'almende" is no longer a common but an individual property, the problem of over-use, which is individually rational but not group rational, does not appear. In principle, this can also be assured by governmental administration of the resource. However, it is not always feasible or acceptable to change a common-pool resource into private or governmental property, nor is it always necessary. Ostrom, Gardner, and Walker [3] investigated how stakeholders deal with common-pool resource situations in a variety of field studies and found that there is great potential to locally solve dilemmas.

There are many different approaches to modeling human behaviour in common-pool resource situations or in any other social dilemma. In this paper we present an approach based on experimental data taken from experiments by Ostrom,

J.S. Sichman and L. Antunes (Eds.): MABS 2005, LNAI 3891, pp. 133–146, 2006.

Gardner, and Walker [4], who conducted several successive experiments with a non-linear common-pool resource setting. These laboratory experiments in the tradition of experimental economics investigate human behaviour in simple controlable situations in order to extract behavioural regularities. The experiments by Ostrom and colleagues provide rich evidence for subject behaviour in non-trivial situations under a variety of institutional settings, including different kinds of sanctioning mechanisms and communication possibilities. Experimental settings and results, as well as an analysis of individual data sets are presented in the next section.

By abstracting even further, agent-based models of these experiments can help to provide a link between data and theory [5, 6, 7]. In case of the model presented here, one objective is to make it possible for humans to identify with agent behaviour. In this objective lies a major difference to Deadman, Schlager, and Gimblett [5], who modelled the same experiments.

Agent behaviour in our model stays as close as possible to the mirco data of the experiments. The agents in our model exhibit a bounded rationality [8]. They have a number of attributes, which constitute the main differences between the agents. They expect other agents to have certain values in these attributes. According to their attribute values they use different heuristics to make decisions and decision changes. They follow an aspiration adaptation procedure [9], using simple search and stopping heuristics. Section 3 introduces briefly the general framework and section 4 gives a description of the model of the experiment.

The model is implemented in a modular way which allows for easy extensions and alterations. A very simple communication mechanism is presented as an extension to the basic design, in section 5. The communication procedure implemented consists of a mechanism to decide on a joint strategy and a mechanism to change agents' expectations of other agents' compliance to the joint strategy.

This paper is concluded by remarks on the objective and usefulness of this modeling approach.

2 Experiment

2.1 Experimental Setting

Ostrom, Gardner, and Walker [4] conducted a series of experiments with a non-linear common-pool resource with varying framings in order to extract dependencies of subject behaviour on institutional settings. Here, only the setting used for the model is presented. Eight subjects are endowed with 25 tokens each. Each subject can invest a number of tokens into the common-pool resource, labeled "'market 2", which has a negative quadratic return function depending on the total amount of tokens invested. All tokens not invested in market 2 are automatically invested in market 1, the outside alternative, which yields a fixed and constant return per token of 5 cent. Market 2 yields a return of 1 cent per output token which is calculated in the following way:

$$Y = 23 \sum x_i - 0.25 (\sum x_i)^2 \tag{1}$$

Y is the total output of market 2 and is divided among the subjects who invested in market 2 in proportion of their share of invested tokens x_i. This game is repeated 20 to 30 times.

The parameters are similar to the discussion in [4]. They were chosen to provide a non-trivial, non-linear environment. The return function of the profit of market 1 plus market 2 is negatively quadratic with a maximum at 36 tokens investment in market 2 (group optimum) and a null at about 109 tokens investment in market 2. With 8 players each endowed with 25 tokens the maximum total investment in market 2 is 200 tokens, which provides a negative return; a drastic overuse of the resource. Group optimum is at 36 tokens total investment in market 2, 4.5 tokens per subject, and zero rent at 72 tokens, 9 tokens per subject. The game theoretic prediction, the symmetric Nash strategy, on the other hand, is to invest 8 tokens, because, assuming all others to behave similarily rational, the ninth token does not yield any more expected return in market 2. For a theoretical discussion see [4, 109ff.]. With these parameters self-organisation is difficult: the group optimum is not at a round number of token investment per person and the focal point of 10 tokens per person constitutes an overuse of the resource.

Communication was introduced in the experiments as either a one-shot communication round after the tenth game or repeated communication starting after the tenth game.

2.2 Experimental Results

Total investment in market 2 follows a pulsing pattern over the course of several games. Often, investment starts at a very high level, yielding negative returns, but drops quickly to profitable levels. After a couple of rounds with token returns in market 2 higher than in market 1 overappropriation occurs again. There seems to be no symmetry in altitude or timing of peaks across experiments. However, the variance in yields decreases over time, yet without stabilisation. In [4] the main variable by which experiments are compared is net yield in percent of maximum. This is also the value presented in Table 1 and Figure 2, that

Table 1. Average net yield as a percentage of maximum in the baseline experiments (no. 35, 39, 40); with one-shot communication (no. 103, 104, 107), with repeated communication (no. 58, 115, 118, 119, 121, 123) and corresponding model runs with 100 parallel games. The values are mean percentages of 5 rounds. The experiment numbers refer to the original numbering by Ostrom e.a.

Rounds	**1-5**	**6-10**	**11-15**	**16-20**	**21+**
Baseline experiments	-43	-12	10	32	-
Model without communication	-44	19	18	26	-
one-shot communication experiments	-41	-13	74	45	43
Model with one-shot communication	-44	19	81	51	45
repeated communication experiments	8	-13	71	79	62
Model with repeated communication	-44	19	80	68	66

compare experimental and model results. For a more detailed discussion of the experimental results see [4, 115ff.].

From the theoretical viewpoint of non-cooperative game theory, face-to-face communication should make no difference in the outcomes of games. However, in this and many other experiments the result is that face-to-face communication does indeed increase the level of cooperation. A reason for this is that subjects are able to assess the trustworthiness of their game partners better when given a chance to see and talk to them [10]. The net yield in percent of maximum was higher after a one-shot communication round, but tended to drop again after a couple of rounds. Repeated communication was even better to enable lasting cooperation. However, overall efficiency was still only 40 to 80% of maximum [4, chapter 7].

2.3 Subject Behaviour

Analysis of individual decisions is very important for the model objective of comprehensible agent behaviour.[1] In this section the results of an analysis of micro data of three baseline experiments are presented.

Subjects received only information about the total investment of the previous round, but not the individual decisions of the other players. So, the only pieces of information that players could use as a basis for further decisions was the total investment and resulting token return of market 2, as well as their own investment decision.

This non-linear situation is actually quite complex with regard to decision heuristics. Of course, players could calculate the group optimal strategy or an individually optimal strategy given the other players' investments. However, since the others will not necessarily repeat their actions in the next round, guessing other players' investment levels is important as well. Five different heuristics are extracted from the data. The first and most important has been mentioned as a rule of thumb in post-experimental questionnaires [4, 121].

Outcome orientation. The most basic decision heuristic is to increase the investment, when market 2 yielded a higher return per token than market 1 and decrease it otherwise.

Anti cyclic decrease and anti cyclic increase. Guessing other players' moves leads to the opposite strategy. If market 2 yields a higher token return than market 1 you can assume others to increase investment and therefore decrease it yourself (anti cyclic decrease) and vice versa (anti cyclic increase).

Trend orientation. Another heuristic depends on the trend of the token return of market 2. If it increased from the second to last round to the last round increase investment and if it decreased decrease investment in the expectation that the trend will continue. Looking at much longer trends does not make much sense because of the pulsing pattern.

Mean orientation. Some try the mean of the previous investment decisions.

[1] I want to thank James Walker for providing me with the individual data of the experiments.

There may be more sophisticated strategies involved, like "setting others up", which are not included in the model. How many of the individual decision changes in the baseline situation can be attributed to these five different heuristics is shown in Table 2. Almost two thirds of all decisions in model and experiment are different from the previous decision. Only these are referred to in Table 2. Anti cyclic behaviour is differentiated into increasing and decreasing behaviour because the logic seems to be different between these two. Deadman, Schlager, and Gimblett [5] explore strategies based on outcome oriented, trend oriented, and mean oriented behaviour. They did not, however, include anti cyclic behaviour.

Table 2. Classification of investment changes due to different reasoning in the baseline experiments and corresponding model run with 100 parallel games, average of rounds 2 to 20. The sum is greater than 100% because a strategy change can be attributed to more than one class.

	Experiments	**Model**
Outcome oriented	65%	62%
Anti cyclic increase	17%	14%
Anti cyclic decrease	17%	15%
Total trend oriented	51%	55%
Possibly mean oriented	12%	8%

The data of experiments with communication is diverse [4, 145ff.]. Most subjects comply to an agreement, some defect a little, sometimes in response to others' defections. Only a few make high defections. With repeated communication, compliance is better than with only one communication round. Reactions on defections are measured reactions rather than grim triggers [4, 199f.].

3 General Framework

The model is implemented in a modular way to allow for easy extensions and alterations. It consists of four distinct parts, the experimenter, the players, the games, and the heuristics. In addition, a builder organises the creation process, so that all parameters, modellers may want to change, are bundled in one builder class. It makes use of factory methods for the creation of players, games, and heuristics in order to make this process re-usable for similar models (see also [11]).

A model is usually scheduled in five phases. In the first phase, the builder creates the experimenter, the games, and the players. In the second phase, players are shuffled and assigned to games. In the third phase, players make their decisions according to the rules of the game and their currently employed heuristic. In the fourth phase the games calculate the outcome according to the players' decisions. Finally, in the fifth phase, data is collected and displayed. In order to model consecutive games, phases two to five are repeated.

Experimenter. The experimenter organizes the timing during a model run. It manages the pool of players, assigns them to games, and shuffles them between games if necessary. It also manages the games, and switches games from active to passive, if necessary. Ideally, the experimenter can be re-used from other experiments, because which games to use in which rounds and whether or not to shuffle the players can be set by the builder.

Players. Player agents have a number of attributes which constitute their personality [11]. The following attributes are implemented, but not all are used in this model:

- Cooperativeness is used to define interest in group utility, usually combined with an efficiency increase [12, 13].
- Conformity is used to define compliance to norms or previously made agreements [14, 15].
- Fairness concerning others is used to define inequality aversion, regarding the utility of other players, but regardless of efficiency concerns [16].
- Fairness concerning me is used to define inequality aversion with respect to the player's own payoff compared to the others' payoffs [16].
- Positive reciprocity is used to define how nice the player reacts on acts perceived as nice [17, 18].
- Negative reciprocity is used to define retaliatory actions in response to acts perceived as hostile [17, 18].
- Risk aversion is used to define ho far an agent refrains from profitable yet risky acts.

The default implementation is that these attributes are independent from each other and take on values between 0 and 1. Players also have expectations about other player's attributes, which are called expected attributes and modelled similar to the player's own attributes. Per default the expected attributes have the same value as the player's attributes. Expectations change according to experiences, while the attributes do not change during a model run, due to the short time horizon.

There is an extensive discussion on *trust* determining experimental subjects' behaviour. Trust is modelled as expected cooperativeness, expected conformity, expected fairness concerning others or expected positive reciprocity, depending on the situation [19, 20, 17].

Players have a pool of heuristics which can be used for a given decision problem. They also can have search heuristics. A search is triggered, when the outcome of a game does not satisfy an agent's aspiration level, which is set at the beginning of a model to a realistic value by the experimenter. The aspiration level is altered by a learning direction mechanism.

Heuristics and attributes are linked with one another in three ways: (1) agent attributes can define which decision heuristic to use, that is, *in which way* a decision is made; (2) the resulting decision of a heuristic can depend on the agent's attributes and expected attributes; and (3) the learning process can depend on the agent's attributes and the employed heuristic. A learning process alters expected attributes and employed heuristics, but not the attributes of an agent.

Games. Decision environments are implemented as games. The abstract superclass of all games provides most of the functionality except those methods that necessarily depend on the rules of the game.

Heuristics. As mentioned above, players have a pool of possible heuristics. This pool is set by the builder, because so far, no mechanism exists that matches heuristics to decision environments automatically. In any case, usually the heuristics and the search of heuristics are the main subject of an exploration.

4 Model

The game implemented for this model sums up the decisions of all players and calculates the outcome according to formula 1. Then the individual returns from market 2 are calculated and assigned to the players as their gains (or losses). Also, individual returns from market 1, in which all remaining tokens are automatically invested are calculated and assigned to the players as gains. The game also lets the player agent know the average decision of the eight players.

In a specific model, agent behaviour is characterised by their heuristics, the initial decision, which depends on the player's attributes, the decision search process, and a learning mechanism. In this model, no learning mechanism has been implemented. This subsection describes the other three aspects for a specific model of the experiments introduced in section 2.

Heuristics. There are 26 possible decisions, corresponding to 0.25 tokens investment in market 2. In linear problems, the range of possible choices has more meaning with respect to COOPERATIVENESS than in this non-linear decision environment. The most cooperative choices in this non-linear setting are 4 or 5 tokens investment. Less is refraining from providing the resource and above is exploitation of the resource.

When an agent is satisfied with the outcome of the decision of the previous round, it simply repeats the decision. If not, a search mechanism is triggered, which incorporates the five ways of reasoning introduced in section 2. This search mechanism is described below. The possible steps, by which a decision is altered are -15, -10, -5, -2, -1, 0, 1, 1 (again), 2, 5, and 10. These numbers result from the data analysis: In the baseline experiments, 73% of all changes correspond to one of these numbers and the mean increase is 3.6 while the mean decrease is 4.1. These decision changes do not, however, cover the few instances, when subjects changed their investment 20 or more tokens and back. This is a strategy that may be called "'setting others up".

Determining the initial decision. In order to reflect the data, an initial distribution of decisions is defined by the builder. Depending on the player agents' cooperativeness the initial decision is either 5, 10, 15 or 25. This is not, of course, the exact distribution in the first games in the baseline experiments. However, in the three baseline experiments analysed, two thirds of the first decisions are either 5, 10, 15, or 25. In later rounds, the effect of prominent numbers [21]

Table 3. The players' cooperativeness determines their initial strategy

Cooperativeness c	Initial strategy	% of agent population	range in experiment
$0.63 < c$	5	37%	3-8
$0.42 < c \leq 0.63$	10	21%	10
$0.17 < c \leq 0.42$	15	25%	13-17
$c \leq 0.17$	25	17%	20-25

is lesser than in the beginning. The higher the cooperativeness, the lesser the initial decision. The thresholds are set to reproduce the initial distribution of these decisions in the three baseline experiments as indicated in Table 3.

Decision search process. As indicated in the description of the subjects' behaviour, there are five different kinds of reasoning involved in changing the decision. Almost all subjects seemed to use these strategies, although with different frequency. Only two of 24 subjects never used anti cyclic behaviour. Therefore, in the model, agents use all five ways of reasoning (see Figure 1). The probabilities for choosing them depend on the situation and the agent's attributes.

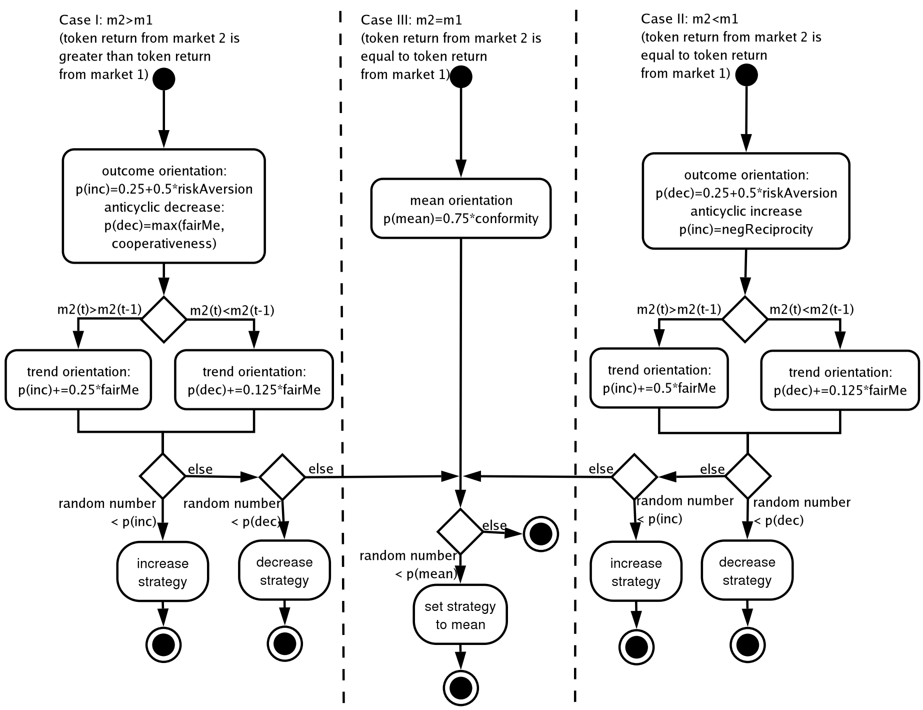


Fig. 1. Strategy changes depend on different probabilities for the five behavioural categories, which depend on the player's attributes and the current situation. (This Figure is an adaptation from [11].)

- High RISK AVERSION leads to a higher probability of outcome orientation;
- high FAIRNESS CONCERNING ME to a higher probability of trend oriented behaviour and anti cyclic decreasing behaviour;
- high NEGATIVE RECIPROCITY to a higher probability of anti cyclic increasing behaviour, because this is the actual defecting behaviour;
- high COOPERATIVENESS to a higher probability of anti cyclic decreasing behaviour, because this can be seen as cooperation;
- finally, high CONFORMITY leads to a higher probability of mean oriented behaviour.

The search process depicted in Figure 1 defines only the direction of the change. The actual change depends on chance and the difference in token returns of markets 1 and 2, as given in Table 4.

Table 4. In outcome oriented reasoning, the height of the change depends on the difference between the token returns of market 2 ($m2$) and market 1 ($m1$). Search strategies are numbered from 0 to 10 according to the following list -15, -10, -5, -2, -1, 0, 1, 1, 2, 5, 10. Values below 0 or above 10 are changed to 0 or 10 respectively. *random* is a random number between 0 and 1.

Reasoning	**Search strategy number**	**Note**
Outcome oriented increase	$5 + 5 * (random + (m2 - m1))$	
Outcome oriented decrease	$5 * (random + (m2 - m1))$	$m2 < m1$
Anticyclic increase	$5 + 5 * random$	
Anticyclic decrease	$5 * random$	

Note, that the search process is only triggered when the aspiration level was not reached. The player's aspiration levels are initially set according to their COOPERATIVENESS within the range 1.2..1.7. Aspiration levels are adapted during the model run. The aspiration adaptation depends on experiences made by agents, as well as their levels of boredom and tiredness. If a decision leads to a satisfactory outcome, the search is stopped, and the decision simply repeated. Each round, in which a repeated decision leads to a satisfactory outcome, the boredom of an agent may be increased, but this was not used in this model. Theoretically, if an agent's boredom reaches a limit, it increases the aspiration level to the average of its previous value and the last outcome and searches again, despite satisfactory outcomes. If an outcome drops below the aspiration level, search is triggered. Each round, in which a search is unsuccessful, that is, a new decision lead to an unsatisfactory outcome, increases tiredness. If an agents tiredness reaches a limit, it decreases the aspiration level to the mean of the previous level and the experience in the previous round. If the search is stopped or started, tiredness and boredom are set to zero.

4.1 Model Results and Validation

For further information on the model, please refer to the model website where the source code is available and parameters are specified.

<www.usf.uos.de/∼eebenhoe/forschung/adaptivetoolbox>.

Moss and Edmonds propose a method of crossvalidation for agent-based models that do not only intend to replicate aggregated data but also the underlying micro processes [22]. One important aspect of model validity is the verification of this micro behaviour. So far, an analysis of individual data was the only source of agent behaviour in the model. It would be a valuable validation process to

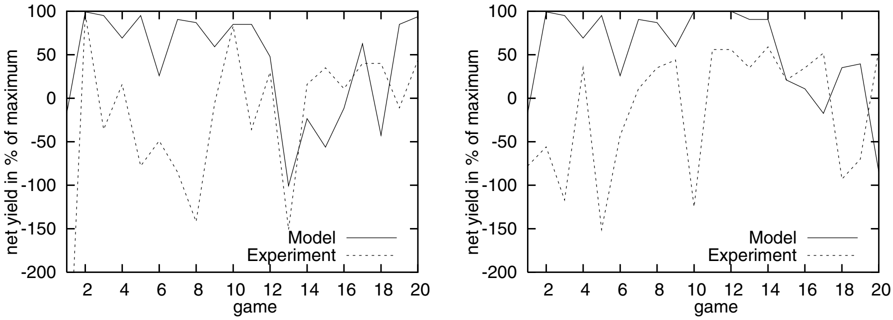


Fig. 2. Results of a selected model run without communication (left) and with one-shot communication (right) and experimental results of experiments 39 (left) and 104 (right)

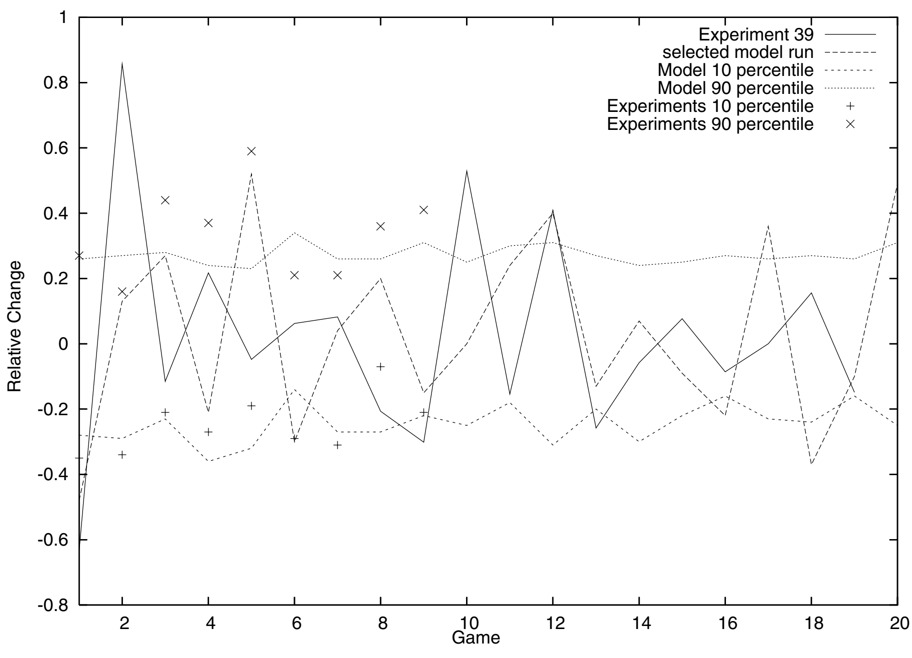


Fig. 3. The relative changes in total investment from one game to the next for the three baseline experiments, as well as 10-percentile and 90%-percentile of 100 parallel model runs and 15 experiments (only for the first 10 games)

learn, whether or not experimental subjects feel that their actions are represented by the decision search process presented here. See [7] for an example of such a process. In our model, agent behaviour depends partly on chance, because it seems reasonable to play this game in an unpredictable way. Another way to validate the micro behaviour is to integrate expert knowledge, in this case that would be the experimenters' knowledge of the subjects' decision making process. This work is in progress.

Because of the nature of the experimental data which has a very high variability and no symmetry across experiments, it is not reasonable to fit this model to a specific experiment. Instead, the qualitative aspects of the data are reproduced. There is a pulsing pattern and an increase in local minimums over time. An example for a model run and an experiment is shown in Figure 2.

In order to show, that the model captures the aggregate data we look at relative changes in total investment from one game to the next. Figure 3 shows the values of the baseline experiment and selected model run from Figure 2 (left handside), as well as the 10-percentile and 90-percentile of 100 models and corresponding percentiles of 15 experiments. The experimental data contains shows only data from games 2 to 10, because after game 10 those experiments enabled communication.

5 Communication

The model is implemented within a framework that allows for easy extensions. As an example for such an extension, a simple communication process was introduced. The implementation of the effects of communication are based on the experiments with communication mentioned in section 2. Here, only an outline of the implementation is given. Introducing communication into the model brings two changes. Agents can agree on a joint strategy and assess the others' COOPERATIVENESS and CONFORMITY better than without communication. Communication is modelled as a three stage process.

1. Each agent announces its *choice decision*, which is collected by a group object and announced to all other players. An agent's choice decision is set initially, like the initial strategy, but follows a slightly different rule: It is 5, if the agents COOPERATIVENESS is greater than 0.5, because then the agent is interested in reaching group optimum, or if its FAIRNESS CONCERNING ME is less than 0.5, because then it is unlikely to defect. In all other cases it is equal to the initial strategy.
2. Each player announces its *acceptance decision*. Again this is collected by the group object and the resulting joint decision is announced to all players. The acceptance decision is the decision that has been mentioned most often as choice, that is still acceptable for the agent. So far, in this model, only their own choice decision is acceptable. However, it is easy to implement other criteria like expectations of higher outcomes, if agreed upon by all, or closeness to their own choice decision. The strategy mentioned most often in

Table 5. Compliance to an agreement depends on acceptance, as well as on expectations of others' compliance, the player's CONFORMITY, and on the attractiveness of a possible defection. In this model, CONFROMITY needs to be greater than 0.5 and expected CONFORMITY greater than 0.75 in order to qualify as high.

Acceptance	**High conformity**	**High expected conformity**	**action taken**
√	√	√	use joint strategy
√	√	–	use joint strategy
√	–	√	use joint strategy
√	–	–	defect a little
–	√	√	use joint strategy
–	√	–	defect a little
–	–	√	defect much if profitable else use joint strategy
–	–	–	defect much if profitable else defect a little

this second stage is given to all agents as *joint decision*, but the use of the joint decision depends on the expectations of the agent (see below).

3. Finally, expected CONFORMITY is adapted to be closer to the actual values of the other agents (see [10, 33f.]). For each agent, the expected CONFORMITY is set to the mean of the previous value and the actual mean CONFORMITY of the other players. Then each player decides whether or not to use the joint decision, depending (1) on acceptance (if it is its own choice strategy), (2) the agent's CONFORMITY, (3) expectations of others' CONFORMITY, and (4) expectations of potential profits from defections (see Table 5). In subsequent games, the expected CONFORMITY is altered by a simple learning direction mechanism. In this way, measured reactions are introduced. If a player wants to comply but expects the others to defect (because they have) it can also defect a little. But if expectations are again in favour of other's compliance it can return to the agreed strategy.

6 Concluding Remarks

In this paper an agent-based model of a non-linear common-pool resource situation is reported, in which the agents exhibit behaviour which is plausible and reflects the micro data. The model reproduces the percentages of five different kinds of decision behaviour as well as the qualitative results of the experiments.

This model is part of a framework for modelling economic experiments, and is therefore, designed in a modular way. This enables modellers to reuse many parts of other models and exchange, for example, only the games and strategies. This design should make it easy to explore different aspects of the same basic model, like different learning procedures, or framing effects. As an example for an extension to the basic framework, a communication process was introduced.

Another extension is the low endowment design, in which the subjects are endowed with 10 tokens each. The model can be changed to reproduce the low endowment experiments quite easily. Game assets are set to 10, but then initial strategies and aspiration levels have to be altered, in order to come up with realistic values. The possible decisions are now only 0 to 10 tokens. The percentages of initial decisions are the same as in the high endowment situation, but instead of 10 tokens investment only 6 are invested, and instead of 15 and 25 the maximum of 10 tokens are invested. Mean aspiration level is reduced by 0.5. With these changes the model yields realistic results for the low endowment situation.

Modelling is an incremental process. This model can be seen as a first step towards representing experimentally observed micro behaviour. Comments and critique on the model can be incorporated to make it easier for subjects to identify with agent behaviour, as is the overall intent of these models.

In addition, the modelling framework should make it possible to implement agent behaviour in *case studies* with behavioural procedures that humans can identify as their own. Experimental data provides a starting point for this objective. The model presented here has not been sufficiently validated, since the source of the heuristics is based on the individual data and not on expert knowledge of people involved in or conducting these experiments. This is not trivial, since subjects may not be willing to make their strategy public. In case studies with a participatory modelling process the actual stakeholders as best experts can help to validate agent behaviour. However, as subjects in experiments, stakeholders are not necessarily willing to reveal their strategies.

References

1. Demsetz, H.: Toward a theory of property rights. American Economic Review **57** (1967) 347
2. Hardin, G.: The tragedy of the commons. Science **162** (1968) 1243–1248
3. Ostrom, E., ed.: Governing the commons: the evolution of institutions for collective action. Cambridge, New York (1990)
4. Ostrom, E., Gardner, R., Walker, J., eds.: Rules, games, and common-pool resources. Ann Arbor, Michigan, The University of Michigan Press (1994)
5. Deadman, P.J., Schlager, E., Gimblett, R.: Simulating Common Pool Resource Management Experiments with Adaptive Agents Employing Alternate Communication Routines. Journal of Artificial Societies and Social Simulation **3** (2000) <http://jasss.soc.surrey.ac.uk/JASSS/3/2/2.html>.
6. Duffy, J.: Learning to speculate: Experiments with artificial and real agents. Journal of Economic Dynamics and Control **25** (2001)
7. Forno, A.D., Merlone, U.: From classroom experiments to computer code. Journal of Artificial Societies and Social Simulation **7** (2004) <http://jasss.soc.surrey.ac.uk/7/3/2.html>.
8. Gigerenzer, G., Selten, R., eds.: Bounded Rationality. The Adaptive Toolbox. Cambridge, Massachusetts: The MIT Press (2001)
9. Selten, R.: What is bounded rationality? [8] chapter 2 13–36
10. Ostrom, E.: Toward a behavioral theory linking trust, reciprocity, and reputation. [23] chapter 2 19–79

11. Ebenhöh, E., Pahl-Wostl, C.: Agent-based modelling with boundedly rational agents. In Rennard, J.P., ed.: Handbook of Research on Nature Inspired Computing for Economy and Management. Idea Group (in press)
12. McClintock, C.G.: Social motivation - a set of propositions. Behavioral Science **17** (1972) 438–454
13. Andreoni, J.: Giving with impure altruism: Applications to charity and ricardian equivalence. Journal of Political Economy **97** (1989) 1447–1458
14. Gerard, H.B., Conolley, E.S.: Conformity. In McClintock, C.G., ed.: Experimental Social Psychology. Holt, Rinehart and Winston, New York (1972) 237–263
15. Simon, H.A.: Models of Bounded Rationality. Volume 3. The MIT Press, Cambridge, Massachusetts (1997)
16. Fehr, E., Schmidt, K.M.: A theory of fairness, competition, and cooperation. The Quarterly Journal of Economics (1999) 817–868
17. Cox, J.C.: How to identify trust and reciprocity. Games and Economic Behavior **46** (2004) 260–281
18. McCabe, K.A., Smith, V.L.: Goodwill accounting and the process of exchange. [8] chapter 18 319–340
19. Nooteboom, B., Six, F., eds.: The Trust Process in Organizations. Edward Elgar, Cheltenham UK, Northampton MA USA (2003)
20. Cook, K.S., Cooper, R.M.: Experimental studies of cooperation, trust, and social exchange. [23] chapter 8 209–244
21. Albers, W.: Prominence Theory as a Tool to Model Boundedly Rational Decisions. [8] chapter 17 297–317
22. Moss, S., Edmonds, B.: Sociology and simulation: Statistical and qualitative cross-validation. American Journal of Sociology **110** (2005) 1095–1131
23. Ostrom, E., Walker, J., eds.: Trust and Reciprocity. Volume VI of Russel Sage Foundation Series on Trust. Russel Sage Foundation, New York (2004)

Tax Compliance in a Simulated Heterogeneous Multi-agent Society

Luis Antunes[1], João Balsa[1], Paulo Urbano[2],
Luis Moniz[2], and Catarina Roseta-Palma[3]

[1] GUESS/LabMAg/Universidade de Lisboa, Portugal
{xarax, jbalsa}@di.fc.ul.pt
[2] LabMAg/Universidade de Lisboa, Portugal
{pub, hal}@di.fc.ul.pt
[3] GUESS/Dinâmia/ISCTE, Lisboa, Portugal
Catarina.Roseta@iscte.pt

Abstract. We consider an individualised approach to agent behaviour in an application to the classical economic problem of tax compliance. Most economic theories consider homogeneous representative agent utilitarian approaches to explain the decision of complying or not with tax payment. However, a heterogeneous and individualised account of decision can be considered to explain certain apparently irrational behaviours. Ideas such as trust and peer perception may have a key influence in individual decisions, and thus transform the global results for society. In this paper, we apply the agent view of rationality to economic decisions and define a territory to be explored by agent technology and social simulations. We conclude that the multi-agent view can provide powerful results which might lead to significant economic policy implications.

1 Introduction

In this paper we extend our views on agent rationality and multi-agent based exploratory simulation to tackle a classic economics problem that continues to defy specialists: tax compliance. The real challenge is not to discover why people evade taxes, but why anyone would pay taxes at all, given the low fines and even lower probabilities of being audited. This is a challenge for a realistic account of rationality, a hard problem for decision theorists. Given that there seem to be some social motivations behind the individual decisions, we propose a multi-agent based exploratory simulation as a way to formulate new hypotheses and conjectures, and perhaps provide some decisive insights into this issue.

The main claim of this paper is that the agent approach to complex systems, enriched by an heterogeneous view of rationality can provide a more realistic picture of the agents true motivations. The idea is not only to improve prediction capabilities (a difficult endeavour because of the increase in complexity brought about by considering autonomous agents, instead of agents with homogeneous

J.S. Sichman and L. Antunes (Eds.): MABS 2005, LNAI 3891, pp. 147–161, 2006.

rationalities), but also to get sharper insights into a hard and complex problem. These insights, conjectures and hypotheses to be tested enhance our deep knowledge of the issue, and may help find new direct and indirect actions to be taken by the policy makers that can be effective both at the global and at the individual levels.

The paper is organised as follows: in the next section, we briefly describe the individual decision model we use for our agents. The relationship between the individual mind (with its own independent, adaptive rationality) and the dynamics of the whole society is very important in this application scenario. The agents' perception of the use of the collected taxes, as well as other social issues as reputation, social trust in both the government and other agents, all have influence in the individual decision. So, in section 3, we present our methodological approach to the exploration of this experimental setting. In section 4 we present some standard economic theories about tax compliance. Section 5 presents a series of increasingly complex models of a tax-aware agent. We end the paper with some preliminary results and some conclusions.

2 An Individualised Approach to Rationality

We have investigated for some time the theoretical underpinnings of rational decision. The standing idea that the choice problem "has already been solved" by utility theory took a severe blow when the economics Nobel Prize was awarded to Kahneman/Smith, in 2002.

The problems with utility theory are well known in the literature, and have been acknowledged in the AI community since the 1950s [20]. However, it is still important to stress that Simon's objection to utility was that it would be in practice impossible to compute. If agents could be instantaneous calculators, utility would provide a *definition* of rationality.

What this view fails to account for is the power of the agent's *will*. Agents will act as they will, depending their own motivations. While Simon coined one of the most important phrases about agent rationality, "people have reasons for what they do" [19], he would still ideally see people as expected utility maximisers. Freedom of choice is the keystone of agent autonomy, but choice is not (only) about calculations or search, rather, it is about volition and motivation. Of course, it remains to be explained why agents choose in the way they do, as their motivations cannot necessarily all be internally generated, some of them are socially influenced individual constructions.

The notion of rationality we will use can be described as *individual*, *situated*, and *multi-varied*. Agent decision can be based not on a single measure of marginal utility but on a multi-dimensional notion of value, which provides a referential for decisions to be taken. This multi-dimensional notion of value provides a referential for decisions to be taken. Then, the consequences of these decisions are assessed, and adjustments are made to the choice mental mechanisms. This choice model is sufficiently adaptive to account for several decision problems, as was extensively tested through simulations [6, 4].

3 Social Simulation and the Micro-macro Link

As we hinted at in the previous sections, the social simulation paradigm might methodologically benefit from taking on the agent stance. In the particular issue of tax compliance, we believe that the agent paradigm will bring along the necessary individualist view of each agent, and so provide social heterogeneity. In the classical sociological accounts (and in particular, in neo-classical econometric theories), practitioners seem to be concentrated in finding a general, all-purpose law that rules everyone's decisions (in this case of complying or evading, and how much to evade). This is obvious in classical examples from the literature, such as Wintrobe and Gërxhani [23] or Andreoni, Erard and Feinstein al. [2].

When social simulation is carried out from the multi-agent systems (MAS) view (and it has been since at least the first SimSoc workshop [15]), the agent view on social interactions becomes foundational. The most important consequence of this is the necessity to take into serious consideration the micro-macro link as described by Conte and Castelfranchi [14]. Roughly described, this means that there is a non-trivial (in fact, complex) interaction between the agents' individual mind and related behaviour and the society dynamics.

In the tax compliance scenario, a simple example can be seen in the trust notion of Wintrobe and Gërxhani [23]. One of the notions of trust the agents possess is trust that the government will provide enough public goods with the collected taxes. According to this model, an agent will evade less the more she trusts the government to fulfil its part of the deal, and vice-versa. The complex view on this mutual dependency goes a little further in detail. What is decisive for the agent's individual decision is not the amount of money the government will invest in the well-being of its governees. Rather, it is the *belief* of the agent about that issue. The realisation of this may lead the government to invest more money in publicity of the popular measures, thus in fact reducing the amount of money available for those measures, and so lowering the overall well-being.

In previous work, we have addressed the issue of methodologically linking the agents' mental structures for decision with the design of the experiment itself [3]. The argument is that by trying to control the agents' and society's behaviour through influencing their motivational system (so respecting the agents' "cognitive and executive autonomy" as Castelfranchi puts it [11]) we obtain a more accurate picture of the real processes that govern individual and global behaviour. The purpose of this move is to be in a better position to interpret results and propose alternative trajectories. By fostering freedom of choice and preserving agent autonomy, we hope that we can influence simulations as little as possible. The ideal situation would be to have completely different people programming the simulations and evaluating their results. But the research cycle for exploratory simulation is complex, and keeping the same people in all stages in fact helps locate the most useful insights (see fig. 1).

We view the work described in this paper as yet another instance where the MAS stance can make a difference by providing a new approach to an old problem. However, the methodologies for MAS, and especially the simulation vein of MAS, can greatly benefit from being confronted with a real issue, and one

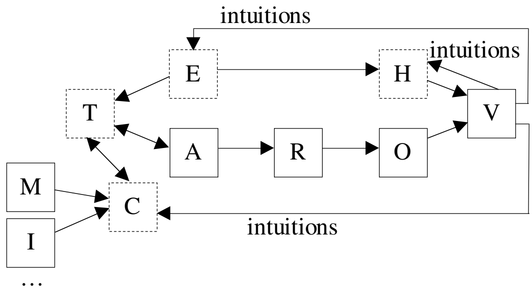


Fig. 1. Exploratory simulation. A theory (T) is being built from a set of conjectures (C), and in terms of the explanations (E) that it can generate, and hypotheses (H) it can produce. Conjectures (C) come out of the current state of the theory (T), and also out of metaphors (M) and intuitions (I) used by the designer. Results (V) of evaluating observations (O) of runs (R) of the program that represents assumptions (A) are used to generate new explanations (E), reformulate the conjectures (C) and hypotheses (H), thus allowing the reformulation of the theory (T). (From [3].)

that has been dealt with in other sciences, with different techniques, assumptions, and/or methods. In the long run, a better (more accurate, more robust, etc.) methodology for MAS experimentation is our ultimate goal, and these cases lead us one step forward.

4 The Economic Theory of Tax Compliance

Tax compliance has been studied in the economic literature from several different and complementary standpoints: public finance, law enforcement, organisational design, labour supply, and ethics (see the survey of Andreoni et al. [2]). Relevant notions in these approaches are equity; efficiency; incidence; effects of penalties; probability of evasion detection; organisational scheme design for taxation, audits and punishment; balance and structure of the labour market, among others.

The most compelling issue for us is the fundamental rational challenge posed to each individual, and how that relates to the overall behaviour of the society. Economists traditionally model individual tax evasion as if the individual is just adding one more risky asset to her household's portfolio [2]. Nevertheless, this theoretical approach fails to explain the behaviour that real societies display: households comply far more than could be predicted in this theory. For instance, in the USA, although fine value (or rate) can be neglected, and even though less than 2% of households were audited, the Internal Revenue Service (IRS) estimates that 91.7% of all income that should have been reported was in fact reported (numbers from 1988-1992-1995, cited from [2]).

In the next subsections, we will present several models of the choice problem each individual faces. Even though these models are very simple to start with, the predictive power they provide through analytical means is very limited, and so they already call for simulation so that the boundaries of their design can be properly explored. Since we intend to provide heterogeneous rationalities for our different agents, and allow agents to change choice settings through interactions and other mechanisms, complexity becomes difficult to overcome. We examine these models as a search for the relevant ingredients to be used later on in more elaborate approaches.

4.1 Standard Theory

Allingham and Sandmo provide a seminal paper about tax evasion [1]. The terminology used there has become standard for the area. A taxpayer has exogenous income y, facing a tax rate t. The amount reported to the government is $x < y$, leaving $z = y - x$ unreported, and paying tax tx. The tax authority does not know the true income y, and has to enforce compliance through a policy of audits and penalties. The model goes on to assume that the enforcement policy is known to the taxpayer and depends on a probability p, with $0 < p < 1$. Further assumptions are that p does not depend on x, and that the tax authority is always able to discover the true value of y. Then, if θ is the penalty to be paid for every unit of income evaded, the cheating taxpayer will have to pay $\theta z + tz$. Given this, and assuming the taxpayer is risk averse, it can be shown that her expected utility if she decides to evade is [2, 23]:

$$(1 - p)u[y(1 - t) + tz] + pu[y(1 - t) - \theta z)]$$

The effect of raising tax t for the "rational" taxpayer decision is not obvious. Raising t exerts contradictory forces in the agent's expected utility. Yitzhaki [24] observed that if the penalty is proportional to the amount of tax evaded (so θtz instead of θz), the model predicts that cheating will be reduced when the tax rate increases. Even if this settles our agents to rest about the decision, it hardly represents a breakthrough for the realism of the model, as noted in [23].

4.2 Public Choice Theory

The basic assumption of the 'public choice' field of economics is that citizens of democratic political jurisdictions perceive a connection between the taxes they pay and the government services they receive [23]. In face of our perception of the micro-macro link and individualised decision, this seems a promising alley to walk when looking for more complex models of taxpaying behaviour. The public choice view implies that every citizen knows that if taxes are reduced then a reduction in public services will follow. Even if the government is seen as a rational utility maximiser, the explanation for citizens to pay their taxes is that they trust the government to deliver the services it has promised. It is nevertheless rational for each individual to free ride, since the level of public services she receives from her own taxes will be negligible.

4.3 Trust Theory

The notion of trust, and related notions such as morality, ethics, categorical imperatives, has gained in recent years an increasing importance in the tax evasion literature. Wintrobe and Gërxhani [23] mention two kinds of trust that influence the tax compliance decision: (i) trust that the government will provide the public goods it is supposed to provide and (ii) trust that fellow tax payers will indeed pay their taxes. Although the focus of Wintrobe and Gërxhani is on the first case, they argue that both types of trust would be identically correlated with the decision to comply or to evade taxes.

In Wintrobe and Gërxhani's trust model it is assumed that only one public good S exists, and all the citizens in this jurisdiction consume the same amount. Each citizen is also assumed to correctly calculate that the tax price to her of a unit of S is p. D is the demand, or marginal valuation curve for the public good. In the authors' analysis, they conclude that if taxes p rise without an accompanying increase in the amount of public good made available, then the propensity to evade taxes will increase. However, in certain conditions, if the rise in taxes is followed by a rise in the level of public goods provided, the propensity to evade will decrease. The role of trust becomes relevant in this model because the citizens will not have a direct perception of the real surplus in public goods deliverance. So, trust takes the role of this perceived surplus: citizens with a larger trust in the government (or that fellow citizens will pay, the argument holds just as well) will more easily support the government, which in turn will return their trust by trying to maximise the sum of citizens' surpluses in terms of public goods.

It is obvious that this trust relationship need not be symmetrical, as is implied by Wintrobe and Gërxhani. For instance, when addressing trust of type (ii), the authors mention (supported by Benjamini and Maital as well as Myles and Naylor [7, 17]) that theoretically there should be a "tipping point" when the number of tax evaders reaches a certain level and cause an epidemics of evasion. In fact, the overall behaviour of societies changes much slower, and all the real numbers show that people comply far more than any theoretical model predicts. It is also naïve to think that the government will always increase the public goods distributed, and only fails to do so because of corruption or lack of knowledge about people's desires. Central administration has costs, and these (as well as inefficiency) have always been neglected in all these models.

A more realistic notion of trust is used by Boadway, Marceau and Mongrain [10]. In this paper, the authors concentrate taxes associated with transactions. This determines that both participants in the transaction must agree whether or not to evade tax, and this joint decision requires that some trust bounds the two participants. Social interactions are modelled as a repeated iterated game, similar to the prisoner's dilemma. Agents are utility maximisers whose behaviour depends heavily on a personal parameter, θ, denoting *tolerance for dishonesty*, and trust is modelled as a kind of generalised *tit-for-tat* observed in all transactions. This being said, the conclusions of this research hold as far as its assumptions are met. Our contention is still that general laws are impossible to find that will govern everyone's mind. We *know* that some people will always pay their taxes, however unfair or nonrational that behaviour may be.

4.4 Extending the Classical Models

The survey paper of Andreoni et al. [2] is very complete and provides the cartography of the field. Although it divides models between "principal-agent" and "game-theoretical," there is a section when some moral and social issues are introduced. However, the effect of these issues on tax compliance literature are described as a "yet largely undeveloped area of research," and the authors add that "little is known or agreed upon about how best to include these effects in a theoretical or empirical analysis of tax compliance."

In light of this view, it is our contention that the agent-based approach with exploratory simulation is the answer to some of the problems raised by the unrealistic assumptions most analyses make. In particular, most models we have encountered carry the burden that all agents must follow the same utility patterns. With agent technology and individualised rationality, we can expect to find all sorts of agents following their own mind. With adaptive choice schemes we will provide agents with the ability to change the way they behave as a reaction to interactions and past events.

Among the proposals for extensions of the models in [2], we found the idea of trust, either in the government or in fellow taxpayers. The idea is to explain the empirically verified non-compliance with the sense of unfairness of the tax payed, either because of little service provided by the central authority, or as a comparison between what the agent pays and what she perceives others to pay. As we have seen, this idea has already been explored in subsequent papers, such as [10, 23]. However, the kind of analyses presented in these papers were also classic, and agent heterogeneity was not explored. Another extension suggested in [2] is the exploration of the moral obligation to be truthful, or the social consequences of being a know cheater. In mental terms, the senses of 'guilt' and 'shame,' respectively. This is another promising path for MAS techniques, since the mechanisms of reputation have been explored in detail for some time now (see the works of Castelfranchi, Conte and Paolucci [12] or Sabater and Sierra [18], for instance). Trust and deception is another area that has been dealt with within MAS for quite some time (see for instance, Castelfranchi and Tan's compilation [13]).

The problem of unrealistic assumptions of models may be a little more difficult to explore, because many of the parameters and constraints come directly from empirical observations or intuitions from practitioners. But even for that matter, the agent-based approach may be better suited and provide a quicker exploration of the space of possibilities, since through simulation the 'bad' choices can be detected and avoided sooner than if real empirical experiments were made.

4.5 Multi-agent Based Simulation

The first Multi-Agent Based Simulation model of income tax evasion was developed by Mittone and Patelli [16]. These authors followed the theoretical work of Myles and Naylor [17] that considers the existence of three categories of tax payers: honest, imitative and free riders. The behaviour of each tax payer category is characterised by a unique utility function. Utility is also influenced by the public sector goods and services supported by tax contributions. This model was programmed using the SWARM package.

Bloomquist [8] developed a Tax Compliance Simulator in NetLogo in which taxpayers are represented as software agents that are defined by a set of characteristics like the income level, the income visible to tax authority, age, perception of enforcement activity, amongst many others. In [9], Bloomquist uses this simulator to conduct an experiment to estimate the deterrent effects of tax compliant alternatives.

5 A Model for Tax-Aware Agents

To commence the exploration of our simulation space we will adopt a very simple model that can fit the requirements of the game theoretical frameworks in the classical literature. However, it is important to stress that ours is not a game theoretical approach. In particular, we are not interested in studying the conditions for reaching some kind of equilibrium, rather we are curious about the overall dynamics of the system. This first approach to the tax-compliance social models will necessarily be very simple. We need to carefully study the dynamics of the simple model before we can start to introduce complexity (namely full agent autonomy and non-rational decision rules, such as moral imperatives).

Thus, in the rest of this section we will introduce an individual agent model that lays the bases for tax compliance decisions. Then we will introduce a model for the central authority, and its rules for tax collection, compliance enforcement, and finally redistribution. We made it a point to start where the classical econometric models start. However, now we depart from those models and delve into the richness of simulation. We propose new variables to illuminate the individual compliance decisions, then propose different categories of agents, each with its own set of motivations and decision procedures. We propose aggregate variables for the central authority and perception functions for the agents to observe them. We end up this section with a description of how agent interactions can determine streaks of behaviour that spread through the population and eventually change the overall dynamics.

5.1 The Ec0 Model

The first model we present is inspired in the classical models of [1], as described in [2]. An agent receives income y of which she will report $x < y$ and omit $z = y - x$. Reported income is subject to a tax t, so every agent will annually pay tx, if she chooses not to comply. In general, the tax may not be linear, so it could be $t(x)$.

Now we postulate a special agent, the central authority, in charge of enforcing compliance. Traditional rational agents would have no reason whatsoever to comply. So, the central authority uses audits and penalties to enforce tax compliance.

In the Ec0 model, we will assume that an agent is audited with probability p and if that happens the true income y will be found out. If the agent is caught evading she will have to pay a fine θ over the evaded amount, in total of θtz.

Note that some of the assumptions made at this point are very drastic:

- Audits are determined by a probability over every return filed. Note that even if probabilities may be a convenient *a posteriori* means for describing the returns examined, no tax agency will exclusively use rolling of dices to decide which return to audit.
- The probability of an agent being audited is independent of the past. This is unrealistic, since the tax agency will always audit an agent more if she has a history of evasion.
- By auditing the returns the tax authority will know the true value y of the income. Even nowadays, in the information era, tax agencies use all sort of data crossing among the millions of documents received.
- The probability of an agent being audited is independent of the probabilities of other agents being audited, and there is no limit for the number of audits to be carried out. As Andreoni at al mention [2], both these assumptions are wrong. More usually, agencies will have a budget which determines the maximum number of audits to be performed. On the other hand, as noted by Boadway et al. [10], agents perform (costly) efforts to avoid (or illude) audits, and this changes the probability that an evader is caught.
- The tax agency decision rule to audit a tax payer (based on probability p) is known by all the agents. This is also unrealistic, as most agencies indeed make enormous efforts to keep those rules secret.
- Another unrealistic assumption that changes the way agents should decide (for utilitarian agents, who compute expected utility) is the notion that when the tax authority finds an evader it will punish her only for the current year. More normally, the discovery of a cheater should induce further investigations on previous years (as well as prompt special attention for subsequent ones). So, the fine should enter the agent's calculations not as θz (or $\theta t z$) but rather as $\theta(z_\tau + z_{\tau-1} + z_{\tau-2} + \ldots)$. On top of that, the event of getting caught probably should influence the future probability of being audited (so increasing $p_{\tau+1}$). Obviously, this makes a substantial difference in both the individual choices and overall dynamics.

5.2 Ec1 and Ec2: Individuality and Adaptability

The first experiments with Ec0 revealed what could be analytically derived from the model: no one would pay taxes unless fines were unrealistically high. This result is typical of classical evasion models.

Our model Ec1 departed from Ec0 by postulating a new parameter to represent the "ethical stance" of the agent and allow for some evolution that could lead us to more realistic results. This ethical attitude is summarised in a number $\epsilon \in [0, 1]$. ϵ is used for the decision rule in such a way that when $\epsilon = 1$ the agent always complies to the tax due, and when $\epsilon = 0$ the agent always evades tax. Of course there are behaviours in between those two. For instance, in one experiment (see below) we used a rule that said the agent would pay when $\epsilon \geq \frac{1}{1+\theta}$, with $\theta > 0$ the fine.

A very important factor for the compliance decision in the classical models is the behaviour of agents in face of risk. Basically, an agent is risk neutral if her utility function is linear, and risk averse if the concavity of her utility function is facing downwards. Even if we are not using utility functions, agents will have to choose, and we can expect those choice functions to share some common ground with classic utility. The classical model assumes agents with decreasing risk aversion. This means that richer agents will more easily gamble with important quantities of money, and so would be more prone to evade. Of course here we have to distinguish between high income agents and wealthy agents. We introduce the quantity w of money an agent possesses. We expect w to be normally distributed, and update w every year by proposing a consumption rate γ to affect income y. To simplify at this early stage of research, we use $w_{\tau+1} = w_\tau + (1-\gamma)y'$, where $y' = (1-t)x + z$, the liquid income of the agent (we follow the notation introduced in section 5.1). Then the agent will be more prone to evade (so have greater ϵ) for greater values of w. For the experiments we used $\epsilon = \frac{w}{w+1}$.

Our Ec1 model credits agents with a limited amount of individuality. However, we need adaptivity, in order to have richer representation of true behaviours. Adding further degrees of complexity to the agents adds on enormously to the complexity of the whole society. So, we used the tactics of adding the mechanisms to allow one parameter to be dynamic, while freezing all others. A different possibility would be to experiment with that same parameter with several distributions, taking empirical studies as a starting point. The problem with that is that we do not know what kind of hidden assumptions were made in the models sustaining those studies.

For our Ec2 model, we proposed a dynamic behaviour for ϵ so had to find a rule for updating its value as a result of the consequences of past decisions [5]. Hence, model Ec2 includes the concept of memory of caught evasions and a factor (δ) for its degradation. If an agent is caught evading taxes, it will necessarily comply that year and will keep complying for some years onwards. So, if an agent is audited and found to be evading, we will set her ϵ to 1. Then, in subsequent years, we will update ϵ with the following rule: $\epsilon_{\tau+1} = \epsilon_\tau - \delta \in [0,1]$.

A more complex behaviour is obtained from relating this ethical attitude with the expectations of surplus received from the central authority in the public choice approach. If an agent observes a personal surplus of So and she was expecting Sp, variation update δ can be written as $1 - \frac{So}{Sp}$. This ensures that if $So > Sp$ then $\delta < 0$ and the agent moves towards paying taxes, and if $So < Sp$ then $\delta > 0$ and the agent moves towards evading.

The distribution of ϵ over the population should be based on psychological evidence, as well as the dynamics of its update. It should be more probable to find individuals with a tendency to always comply, and also individuals with a tendency to always evade. Even if some individual realises that he is paying too much and decides to start evading, it is not clear that this change would be gradual, as our model proposes. A possible credible distribution for ϵ could be similar to an inverted Normal, with lots of agents always complying (or slightly hesitating to do so) and lots of others always evading (or slightly hesitating to do so). Our first experiments used a Uniform distribution, and we are currently undertaking experiments with other distributions.

5.3 Ec3*: Introducing Sociality

So far, the decision problem each agent has to face is an individual one. In Ec3* we introduce sociality: agents are aware of other agents and each individual decision depends on perceptions about the society as a whole. In subsequent models we will introduce interactions, and other forms of more complex social behaviours that contribute to overall social dynamics.

Following the idea of keeping experiments simple (in the path of constructing a "broad but shallow" agent design [21]), we introduce in the agent's design a perception of the level of overall fraud, $\Phi = \frac{\sum_{i \in Agents} z_i}{\sum_{i \in Agents} y_i}$. The agent's attitude towards this quantity will represent her level of comfort in face of unfairness. So we introduce $\phi \in [0, 1]$, the threshold of overall fraud admissibility for each agent. If $\Phi > \phi$ then the agent evades tax.

A slightly more complex model would include $\xi \in [0, 1]$ to represent the tolerance of agent in presence of global fraud. So, if $\xi = 0$ the agent ignores Φ and ϕ and complies, if $\xi = 1$ the agent follows the above rule.

Another interesting rule to explore is the local perception of fraud instead of the global perception. For instance, an agent will evade if a significant percentage of her neighbours evades. Quantity Φ is so changed to $\Phi_a = \frac{\sum_{i \in Neigh(a)} z_i}{\sum_{i \in Neigh(a)} y_i}$. Experimentation with these models is currently being conducted, and we expect that the crossing of its results with empirical data helps us make the necessary design options among the possibilities here outlined.

6 Preliminary Results

The results reported in this section were obtained conducting the described experiments using version 2.0.2 of the NetLogo framework [22]. NetLogo is a programmable modeling environment for simulating natural and social phenomena. It is particularly well suited for modeling complex systems developing over time.

In all experiments we consider two types of parameters: *internal* and *global*. Internal parameters are the ones that distinguish agents from one another (for instance, the agents' annual income). There are two types of global parameters: they can represent values that are equal for all agents (the tax rate, in the preliminary experiments) or values that the individual agents don't know at all (the probability of being inspected).

At this stage of research, we are still fine tuning some of the models and getting insights on how the several variables influence the system dynamics. So, in this section we will only hint at the kind of results we obtained, and present a brief analysis.

To test our model Ec2, we investigated how we could reduce the number of evading agents with the use of the memory effect on getting caught. So, for

Table 1. Percentage of evaders for several values of θ and p over δ

	θ	0.25			0.5		
	p	0.01	0.05	0.1	0.01	0.05	0.1
δ	0.01	.83	.51	.34	.75	.37	.23
	0.02	.91	.67	.51	.85	.55	.38
	0.03	.93	.74	.59	.89	.63	.46
	0.04	.95	.80	.67	.92	.69	.53
	0.05	.96	.83	.71	.93	.74	.59
	0.1	.98	.87	.77	.97	.84	.71
	0.2	.99	.91	.83	.99	.91	.83

reasonable (from the literature) values of the tax rate ($t = 0.30$), fine ($\theta = 0.25, 0.5$), and of number of audits to perform a year ($p = 0.01, 0.05, 0.1$), we varied the tendency for evading (the number δ to be subtracted from ethical attitude ϵ) from 0.01 to 0.05, and then 0.1 and 0.2. The results are shown in table 1. In every cell we have the mean percentage of evading agents in a population of 500 agents, with a normal distribution for the income and initial uniform ϵ subsequently updated as described in Ec2 above. We performed some dozens of experiments, and the numbers here presented are typical of what happened. Some stationary equilibrium was always reached, with some stochastic variability. Other measures were collected (total tax collected for redistribution, variability of ϵ, mean agent wealth, etc.) but their analyses are not completed yet.

As δ gets closer to zero, the memory effect of getting caught is stronger. So, agents will keep a good ethical attitude longer, and this gets reinforced by getting caught again. On the other hand, if agents do not evade taxes, their attitude leans towards evasion, at a pace set by δ. Looking at the table, we can see that with audit probabilities of 10%, we can still get relatively low numbers of evaders by keeping memory degradation slow. Without the memory effect, we could expect a mean of evaders approximately equal to 90%. These positive results are stressed by increasing the fine to 50% of the evaded amount. For instance, with a 1% probability of inspection we can have 25% of compliance, which is remarkable. We believe we can still improve these results by associating memory with the corresponding years, instead of considering the compliance decision of one year independent of all others.

7 Concluding Remarks

The main point of this paper is to argue that multi-agent systems concepts and techniques are a step forward in social science problems that have resisted more traditional approaches. The classical problem of individual tax compliance, as well as the problem of determining the correct tax enforcement policy, have constituted for decades a challenge for economists, sociologists and social psychologists. From our point of view, it is also an interesting problem for MAS practitioners, since it presents a clear case where the limits of situated rationality are put to test, and the neo-classical economics approach of maximising expected utility remains wanting in face of the empirical results available. Tax compliance is also a great issue to test out agent based simulation methodologies and techniques, and to perform exploratory simulations that can help tackle the hot questions themselves, while gaining in experience and improving the necessary methodologies for experimentation with self-motivated agents.

In this paper we define the borders of this region we want to explore, and enumerate some of the difficulties already known in the literature, as well as criticize some of the assumptions made in other approaches. We then propose an approach based on multi-agent based simulation and describe agent models and experimental settings to be used. We already performed some experiments and the results are encouraging. However, this research is only beginning, and the complexity of the tasks at hand recommend prudence.

The promise of the first exploratory experiments, and perhaps the only true conclusion of this paper, is that the approach will deliver insights into the problem that were not available previously. Agent heterogeneity and individuality will provide a more realist account of the rational decisions that determine the overall behaviour of the society. And, above all, agent technology and exploratory simulation will (and already did) allow for the rehearsal of mechanisms to try out different design scenarios. The idea is to increasingly deepen the mechanisms and decision modules, in a plug-and-play manner, to provide enhanced adaptability to the world conditions, as well as allow for external (agent to agent, for instance by goal-adoption) and internal interactions (interacting, and possibly conflicting parts of the agent's mind).

Future work will first deal with the validation of empirical data, allowing us to settle on an accepted paradigmatic agent. Then, we will define this agent and continue exploration, by stretching and deepening some of its mechanisms. We will also look carefully into the measures to be taken from both experiments and real world, and the methods to validate the simulation results. Another important issue is to look into the role of the central authority in what concerns policy decisions. As well recognised in the literature, tax compliance (theory and policy) is a fundamental matter for today's countries, especially those in transition to democracy. We believe that more realistic multi-agent decision models will provide a helpful contribution in this policy area.

References

1. Michael G. Allingham and Agnar Sandmo. Income tax evasion: A theoretical analysis. *Journal of Public Economics*, 1(3/4):323–338, 1972.
2. James Andreoni, Brian Erard, and Jonathan Feinstein. Tax compliance. *Journal of Economic Literature*, 36(2), June 1998.
3. Luis Antunes and Helder Coelho. On how to conduct experiments with self-motivated agents (revised). In Mario Paolucci Gabriela Lindemann, Daniel Moldt, editor, *Regulated Agent-Based Social Systems: First International Workshop, RASTA 2002, Bologna, Italy, July 16, 2002, Revised Selected and Invited Papers*, volume 2934 of *Lecture Notes in Artificial Intelligence*, pages 31–47. Springer-Verlag, 2004.

4. Luis Antunes, João Faria, and Helder Coelho. Improving choice mechanisms within the BVG architecture. In Yves Lésperance and Cristiano Castelfranchi, editors, *Intelligent Agents VII, Proc. of ATAL'00*, volume 1986 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2000.
5. Luis Antunes, João Faria, and Helder Coelho. Choice: the key for autonomy. In Pavel Bradzil and Alípio Jorge, editors, *Proceedings of EPIA 2001*, Lecture Notes in Artificial Intelligence. Springer-Verlag, 2001.
6. Luis Antunes, Leonel Nóbrega, and Helder Coelho. BVG choice in Axelrods tributes model (revised). In Jaime Simão Sichman, François Bousquet, and Paul Davidsson, editors, *Multi-Agent Based Simulation II*, volume 2581 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2003. Third Internacional Workshop, MABS 2002, Bologna, Italy, July 2002.
7. Y. Benjamini and S. Maital. Optimal tax evasion and optimal tax evasion policy: behavioral aspects. In W. Gaertner and A. Wenig, editors, *The Economics of the Shadow Economy*, pages 245–264. Springer-Verlag, Berlin, 1985.
8. Kim M. Bloomquist. Modeling taxpayers' response to compliance improvement alternatives. Extended abstract at the NAACSOS Conference, June 2004.
9. Kim M. Bloomquist. Multi-agent based simulation of the deterrent effects of taxpayer audits. 97th Annual Conference of the National Tax Association, November 2004.
10. Robin Boadway, Nicolas Marceau, and Steeve Mongrain. Tax evasion and trust. Technical Report 104, Center for Research on Economic Fluctuations and Employment, Université du Québec à Montréal, February 2000.
11. Cristiano Castelfranchi. Guarantees for autonomy in cognitive agent architecture. In Michael Wooldridge and Nick Jennings, editors, *Intelligent Agents: agent theories, architectures, and languages, Proc. of ATAL'94*, volume 890 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 1995.
12. Cristiano Castelfranchi, Rosaria Conte, and Mario Paolucci. Normative reputation and the costs of compliance. *Journal of Artificial Societies and Social Simulation*, 1(3), 1998. http://www.soc.surrey.ac.uk/JASSS/1/3/3.html.
13. Cristiano Castelfranchi and Yao-Hua Tan, editors. *Trust and Deception in Virtual Societies*. Kluwer Academic Publisher, Dordrecht, 2001.
14. Rosaria Conte and Cristiano Castelfranchi. *Cognitive and Social Action*. UCL Press, London, 1995.
15. Nigel Gilbert and Jim Doran, editors. *Simulating Societies: the computer simulation of social phenomena*. UCL Press, London, 1994. Proceedings of SimSoc 1992.
16. L. Mittone and P. Patelli. Imitative behaviour in tax evasion. In B. Stefansson and F. Luna, editors, *Economic Modelling with Swarm*, pages 133–158. Kluwer, Amsterdam, 2000.
17. Gareth D. Myles and Robin A. Naylor. A model of tax evasion with group conformity and social customs. *European Journal of Political Economy*, 12(1):49–66, 1996. available at http://ideas.repec.org/a/eee/poleco/v12y1996i1p49-66.html.
18. Jordi Sabater and Carles Sierra. Reputation and social network analysis in multi-agent systems. In *The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002, July 15-19, 2002, Bologna, Italy, Proceedings*, pages 475–482, 2002.
19. Herbert Simon. Rationality in psychology and economics. In R. M. Hogarth and M. W. Reder, editors, *Rational choice: the Contrast Between Economics and Psychology*. Univ. of Chicago Press, 1987.

20. Herbert A. Simon. A behavioral model of rational choice. *Quarterly Journal of Economics*, 69:99–118, February 1955.
21. Aaron Sloman. Prospects for AI as the general science of intelligence. In *Prospects for Artificial Intelligence - Proceedings of AISB'93*. IOS Press, 1993.
22. U. Wilensky. NetLogo, 1999. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL. http://ccl.northwestern.edu/netlogo/.
23. Ronald Wintrobe and Klarita Gërxhani. Tax evasion and trust: a comparative analysis. In *Proceedings of the Annual Meeting of the European Public Choice Society – EPCS 2004*, 2004.
24. Shlomo Yizthaki. A note on income tax evasion: A theoretical analysis. *Journal of Public Economics*, 3(2):201–202, 1974.

Agent Transport Simulation for Dynamic Peer-to-Peer Networks

Evan A. Sultanik, Maxim D. Peysakhov, and William C. Regli

Department of Computer Science, Drexel University, Philadelphia, PA, USA

Abstract. This paper introduces MATES: the Macro Agent Transport Event-based Simulator. MATES is a free, open-source simulator intended for efficient comparison of application layer agent algorithms. Existing simulators, both from the networking and artificial intelligence communities, are not adequate to efficiently model the domain of mobile agent systems running atop dynamic, peer-to-peer networks. MATES bridges the gap between the two disciplines by providing macroscopic models for low-layer network processes, allowing for more processing time to be devoted to the agent behavior model. The three principle contributions of this paper are an introduction to the simulator, validation of MATES' models using data from a live wireless network, and explication of a technique for improving simulator transparency through preservation of agent state.

1 Introduction

In developing the Secure Wireless Agent Testbed [1], we have experienced first-hand numerous problems that arise from working with agents on live wireless networks. Hardware failures and system misconfiguration can cost valuable research time. Although producing agent-based technologies for a functional wireless network has always been our goal, we have recognized the need for a simulated environment in which to compare agent-based algorithms and perform controlled experiments. To address this need we have developed the Macro Agent Transport Event-based Simulator (*MATES*). MATES is an application-layer simulator created to investigate the behavior of distributed agent-based systems running atop dynamic peer-to-peer and mobile ad hoc networks (MANET). Given such a goal, MATES was implemented *not* to simulate any specific agent framework, but to provide a generic, easily extendible environment for mobile agent-based system testing.

In the context of this paper, a "*mobile agent*" refers to a program capable of halting its execution and migrating to another *host*, at which the program will continue execution. A "*host*," therefore, is any entity on the network capable of receiving and executing a program. A dynamic peer-to-peer network is a group of hosts that may not always be fully-connected, and whose connectivity may be in constant flux. MANET are a specific type of dynamic peer-to-peer network in which hosts may move; the network topology is determined by the hosts' spatial interrelationships.

J.S. Sichman and L. Antunes (Eds.): MABS 2005, LNAI 3891, pp. 162–173, 2006.

The rest of this paper is organized as follows: we present our motivation for developing MATES in §2, followed by a description of MATES' architecture in §3. §4 presents implementation details for MATES. Empirical validation of the scalability of MATES is presented in §5. Examples of how MATES has been used and validation of its models are presented in §6. We present limitations of the current implementation, as well as mention possible improvements, in §7.

2 Motivation and Design Goals

The domain of mobile agency on ad hoc networks is unique, in that it falls between the fields of artificial intelligence and networking. There already exist many simulators from each field, such as MASON [2], for agent simulation, and the almost ubiquitous NS-2 [3] for network simulation.

The authors considered extending an existing simulator for use with dynamic peer-to-peer networks, however none allowed for their specific design goals. The first design goal was that the simulator should only model the network at a macroscopic level, allowing for computation time to be diverted to the agent model. Secondly, developing agents for the simulator should be analogous to development for a real-world agent architecture, such as EMAA [4] or COUGAAR [5].

The first design goal eliminated virtually all network simulators, as their purpose is to model low level interaction (even down to the physical layer). Network simulators are primarily used to compare low-layer processes, such as routing algorithms. As such, most network simulators do not provide sufficient hooks for implementing application-layer protocols. Modeling an agent system at such a resolution has proven to create a fair amount of overhead. Therefore, MATES was created to approximate the low-layer processes to divert more resources to the agent model. This concession of accuracy mitigates the problem of efficiency; further analysis is provided in §6.

A huge disparity between agent simulators and actual multi-agent systems is their scheduling model: the means by which agents are allotted processing time. Real-world multi-agent systems (especially mobile agent systems) almost always provide each agent with its own thread of execution, allowing for agents to run concurrently. In these systems, agents need not compartmentalize their execution into blocks over which the system will iterate. Likewise, agents need not schedule future execution times with the agent architecture. Instead, since each agent is encapsulated by a thread, the operating system or virtual machine can allot processing time. Most all agent simulators take an opposite approach. Simulators tend to provide each agent with a hook that gets called once every simulator iteration. This makes modeling processing constraints unintuitive. For example, it is not immediately clear how one would model a network of heterogeneous hosts, each having drastically different computational power. An agent currently hosted on a PDA might only have a tenth the computational capability of an agent hosted on a laptop. MATES' solution to this problem is presented in §3.2.

3 Architecture

MATES is based upon four core models used for simulation: *Host Mobility*, *Link Connectivity*, *Transport*, and *Agent Behavior*. Each model can be thought of as a conglomeration or approximation of an associated "block" of the OSI model [6]. During every cycle of the simulator, each of the models is successively applied to the domain, as in Figure 1.

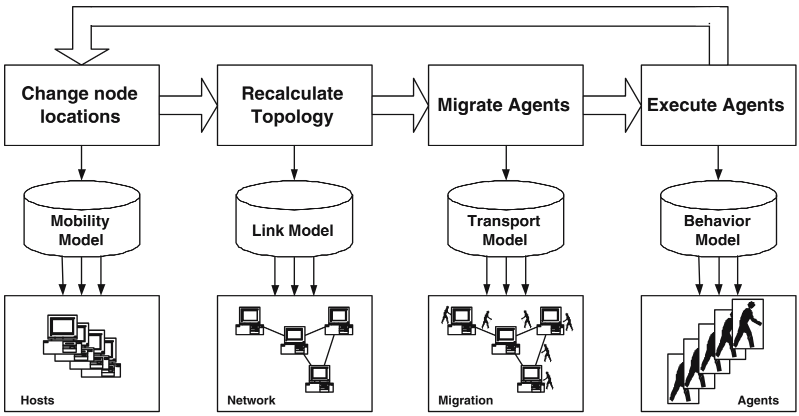


Fig. 1. The simulator cycle

3.1 Hosts and Agents

MATES models two primary entities: *hosts* and *agents*. Unlike some agent architectures, such as EMAA, MATES does not provide a thread of execution for servers. Instead, static agents (i.e. agents that never migrate) can be used.

Each agent can query its current host for the following percepts:

- handles to any *other* agents at the current host;
- geographic location of the current host; and
- network addresses of the neighboring hosts.

It is also important to note that agents do not exist on any host while in transit between hosts. Furthermore, agent migration can fail, for example, if the source and destination hosts move out of range mid-transmit, or if a timeout occurs. In the event of a migration failure, simulation parameters can dictate whether an agent returns to the host from which it was sent or dies. The latter case can be useful for modeling network packets as agents [7].

3.2 Simulated Time

Time in MATES is modeled using *iterations* that represent simulated time quanta. The simulator iterates over the quanta, allotting processing time to each of the hosts. Hosts, in turn, may divide their allotted time to any agents they are hosting. Similar approaches are taken by most other agent simulators.

MATES differentiates its approach by additionally providing each agent with a separate thread of execution. This does not mean that agents execute concurrently; the threads are used to allow for agents to *suspend* their execution while waiting on blocking operations, such as sleeping and migrating between hosts. When the blocking function is complete, the agent's thread may then be started again, continuing execution at the same point at which it suspended. The flow of control while simulating the behavioral model is pictured in Figure 2. The approach of preserving agent execution state throughout an entire simulation is similar to that of Java in Simulated Time [8].

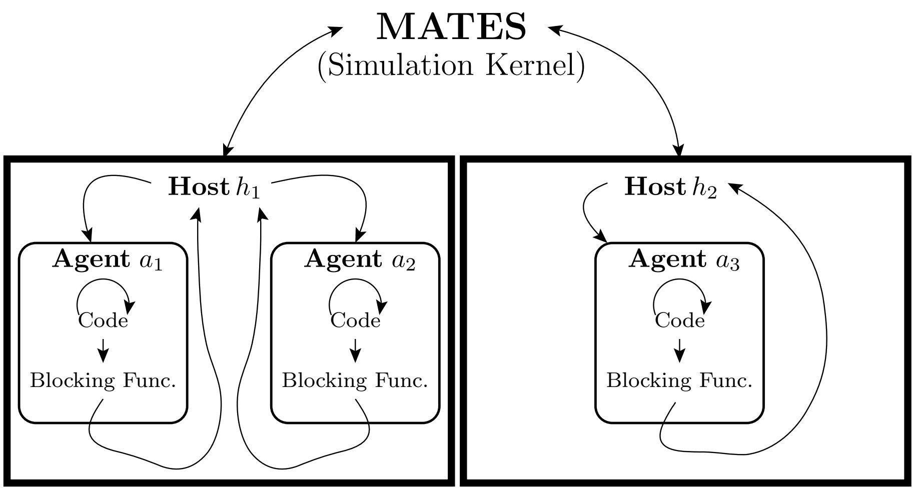


Fig. 2. Flow of control of the behavioral model in a simulation of two hosts and three agents. MATES first allocates execution time to h_1, which sub-allocates time to its agents a_1 and a_2. When each agent reaches a blocking function, control returns to the simulation kernel. When h_1 is finished, h_2 will be given execution time. Finally, h_2 allocates execution time to its agent a_3.

Suspension of agent execution while preserving execution state allows for a familiar agent development environment and greatly increases the transparency of the simulator. This imparts a sense of "continued execution" from the point of view of the programmer. For example, consider Algorithm 1. This algorithm describes an agent that randomly walks the network, printing "Hello World" at each host visited. Note that lines 3 and 5 of the code are blocking. Ideally, one would like other agents on a host to acquire processing time once the currently-executing agent goes to sleep or migrates.

If implemented in a traditional agent simulator, the Hello-World-Agent would have a function that would be called once every simulator iteration. Therefore, to have the same functionality as in Algorithm 1, the iteration function would require some sort of finite state machine to keep track of the agent's state. In MATES, however, each agent's main execution function is called only

Algorithm 1. Hello-World-Agent

```
while True do
    Print("Hello World")
    Sleep(1000)
    N ← Get-Neighboring-Hosts()
    Migrate-To-Host(Random(N))
end while
```

once: when the agent is instantiated. In MATES, one could implement the Hello-World-Agent's functionality in the exact same procedural form as in Algorithm 1. MATES accommodates for this by halting an agent's thread when it reaches a blocking function (i.e. a sleep, migration, or yield).

3.3 Host Mobility Model

Every host has a *mobility model* that dictates the way in which it moves. During every successive quanta of simulated time, each host's mobility model can dictate its location anywhere within the bounds of the simulated environment (which are a simulation parameter). In most cases, mobility models will move the host to a position adjacent to its current, however this is not a requirement.

3.4 Link Connectivity Model

The simulation itself has a *link connectivity model* that defines the conditions under which two hosts have a connection. This allows for simulation of both static and ad hoc networks. In the former case, the link connectivity model is basically a lookup table for connections. In the latter, link connectivity is a function of hosts' locations. The link connectivity model also determines links' quality: a metric roughly corresponding to signal strength in a wireless network.

3.5 Data Transport Model

The simulation also has a *data transport model* that defines the amount of time required for an entity to be sent over a specific link. Here, "time" is actually referring to simulator quanta. This will usually be a function of the link quality and the size of the entity.

3.6 Routing

Agents are assumed to route themselves, in the *Active Networking* paradigm [7]. However, it is possible to implement a routing protocol in the simulator. This can be accomplished by having a static agent on each host responsible for carrying out the routing tasks. When an agent needs to know the next hop in a route, it will query its host for a handle to the "routing agent," and query that agent for the route table.

4 Implementation

MATES is free, open-source, and is hosted at `http://mates.sourceforge.net/`. The current version of MATES has been implemented in Java, chosen primarily for its ease of its class polymorphism, reflection, and extension. Also, many of the current leading agent architectures are implemented in Java [4, 5]. A screen shot of our implementation of MATES is provided in Figure 3. The implementation provides built-in host mobility, link connectivity, and data transport models, however each of these can be overridden.

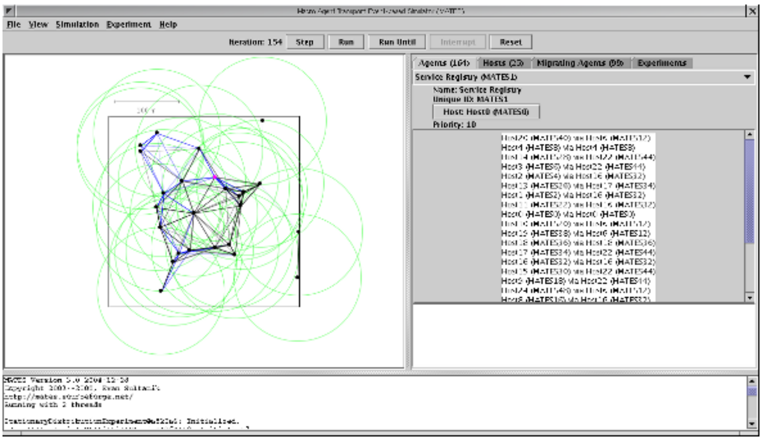


Fig. 3. Screen shot of MATES' GUI, simulating a mobile ad hoc network of 25 hosts

Variations of the *Random Walk Mobility Model* and *City Section Mobility Model* [9] have been implemented. The host mobility model is passed a handle to the link connectivity model during host placement. This enables the host mobility model to move hosts in such a way to preserve a certain topology. For example, one might never want the topology to be disconnected. Therefore, the host mobility model can predetermine if a specific host-repositioning will disconnect the network.

The default link connectivity model emulates the *exact connectivity* model for ad hoc network graph generation [8]. Connections are determined by the Euclidean distance between hosts. Each host has a default radio range of 300 meters, which can vary on a host-to-host basis. Also note that, since hosts can have varying radio ranges, the resulting network topology is a directed graph. This means the fact that host X can "hear" host Y *does not* imply that host Y can hear host X.

The default data transport model dictates that transit times are calculated with an inverse linear relationship to link quality. Therefore, agent transit times have an exponential relationship to the Euclidean distance between hosts. By default, the maximum transit time is a constant number of quanta; all traffic takes the same amount of time to transmit. However, each agent may implement an interface such that it can dynamically define its own transit time, allowing for simulation of agents of different size.

5 Scalability

The two primary parameters that dictate the MATES' computation and memory complexities are the number of hosts and the number of agents in simulation. We profile these complexities by varying the parameters over a simulation of 500 quanta. The hosts moved according to the City Section Mobility Model [9] and each agent performed a random walk on the network. The domain was restricted to a 300 meter square, and hosts' radio ranges were set to 100 meters. The data were collected on a computer with dual AMD Opteron 240 processors, 1GB of RAM, and running Linux kernel `2.6.7-gentoo-r9`. `Blackdown-1.4.2-rc1` was used as the Java runtime environment.

The results for the computational complexity of running the experiment are given in Figure 4. The times recorded in the figure are the amount of CPU time that elapsed between the start and end of the experiment; the time required for MATES to initialize was not included. The data indicate that computation time scales linearly with respect to the number of agents in the simulation, and scales polynomially with respect to the number of hosts in the simulation.

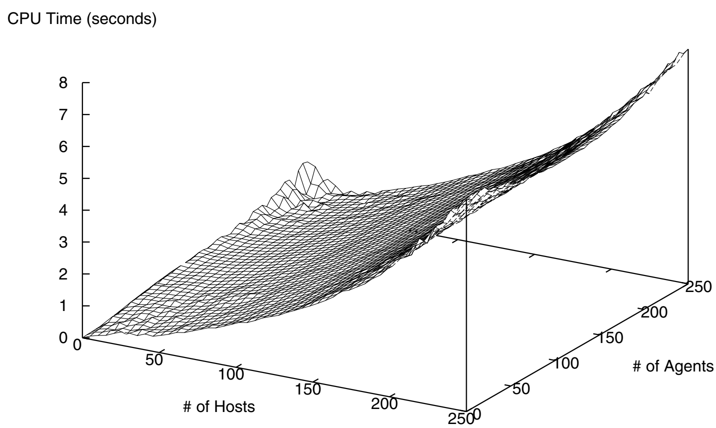


Fig. 4. Computation time for simulating 500 quanta as a function of the number of agents and the number of hosts

The results for the memory-space complexity of running the experiment are given in Figure 5. These data are inherently error-prone, as the Java virtual machine will often allocate more memory than it actually requires. The data indicate that memory usage scales linearly with respect to the number of agents in the simulation, and scales polynomially with respect to the number of hosts in the simulation.

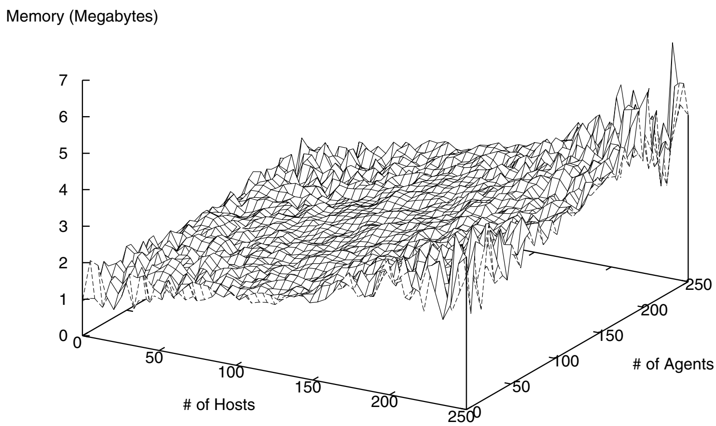


Fig. 5. Memory required for simulating 500 quanta as a function of the number of agents and the number of hosts

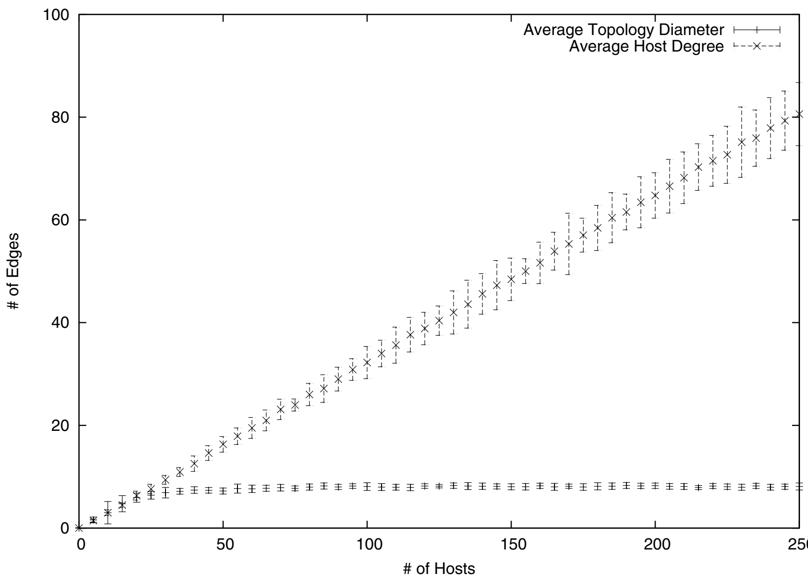


Fig. 6. Topological statistics recorded over the experiments

Intuitively, the complexity of the simulation is not only dependent on the number of hosts, but also the topology of the network. For example, a completely disconnected network (with no edges) will simulate much faster than a completely connected network. Therefore, we also investigate the correlation between the complexity of the simulation and the edge density. Figure 6 shows the average topology diameter and average host degree over the experiments as a function of the number of hosts. Since the mobility of the hosts

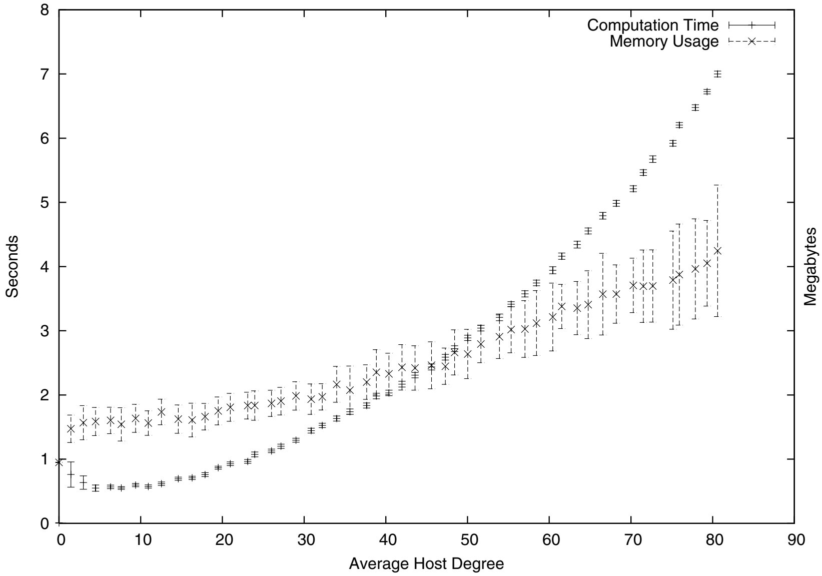


Fig. 7. Computation time and memory usage as a function of average host degree. Note that both values are highly correlated to host degree

are constrained to a 300 meter square, adding more hosts linearly increases the average host degree. Likewise, once the square becomes saturated with hosts (which occurs at about 45 hosts) the average topology diameter becomes constant.

Figure 7 shows the computation time and memory usage for the experiments with respect to the average host degree. Both computation time and memory usage are highly correlated, however the computation time data have very low variance. This indicates that the computational complexity is much higher with respect to the number of edges in the network than with respect to the number of agents.

6 Examples and Validation

MATES has already been used for experimentation in various areas of both artificial intelligence and networks research. In these experiments, simulated data from MATES has shown to correlate strongly to equivalent experiments conducted on live MANET. Networking topics studied with MATES include MANET power management and localization (based on partial GPS information and signal strength). The remainder of this section outlines artificial intelligence research conducted using MATES.

Agent Population Management. One of the first experiments conducted using MATES entailed analysis of an ant algorithm for estimating and managing the optimal number and location of services in a multi-agent system deployed over a MANET [10]. Specifically, service availability[1] was measured under variation of:

- number of hosts on the network;
- link models; and
- network mobility models.

The experiments were conducted both with and without ants managing services. MATES' random number generation architecture allowed for the exact same sequence of host movement during both control and variable experiments.

The same experiments were then conducted on the Secure Wireless Agent Testbed [1], running on iPAQ PDAs. The agents were implemented using Lockheed Martin's Extendible Mobile Agent Architecture (EMAA) [4]. MATES' "continued execution" architecture allowed for agent code to be ported almost directly from MATES to EMAA. As detailed in [10], the experimental results from MATES mirrored the real-world empirical results from the SWAT almost exactly.

Agent Ecosystems. MATES has also been used to simulate agent ecosystems, in which the supply and demand for virtual food is used to optimize the stability of an agent system [11, 12]. In other words,

- Agents collect food as a reward for completing a task.
- Agents consume food regularly to stay alive.
- Agents that exhaust their food supply die.
- An abundance of food causes new agents to spawn.

A formal model of this system was created, and was validated under MATES.

Service-Based Computing. MATES is one of the few environments available in which one is able to extensively test service-based computing and the agent paradigm in the domain of dynamic networks. Because of this, MATES was chosen to simulate and study the feasibility of service-based computing on disruption and delay-prone networks [13]. Multiple service discovery methods were compared on simulated satellite relay, interplanetary, and mobile ad-hoc networks. It was shown that, even in optimal network conditions, effective agents must perceive, reason, and act on network state.

7 Limitations

A major limitation of MATES, in its current implementation, is its atomic representation of time. This means that all user-created simulation components (i.e. agents) must agree on a mapping from quanta to "simulated time."

MATES was not designed for large networks, and as such does not make an effort to aggregate or parallelize homogeneous entities and tasks. With that said, MATES has exponential computational complexity with respect to the number of agents and hosts, with memory usage scaling linearly.

[1] "Service availability" is a metric for a host's ability to efficiently utilize a service on the network.

8 Conclusions

This paper has introduced MATES: the Macro Agent Transport Event-based Simulator. A "continued execution" architecture was also proposed, providing a means for simulators to increase transparency and appear akin to development for real-world agent systems. The validity of MATES' models was then confirmed through correlation between simulated experiments and live, empirical data.

Dynamic, peer-to-peer networks and agency are both heavily researched areas of computer science. As these types of networks become more prevalent, efficient distributed algorithms will be required; the need for further research in this area has already been established [14]. Therefore, a system for testing and comparing these algorithms before deployment is a necessity. We believe MATES, or a system like it, is a step toward achieving this.

Acknowledgments

Our thanks to Joseph Kopena, Robert Lass, and Andrew Mroczkowski for their use of and comments on the system.

References

1. Sultanik, E., Artz, D., Anderson, G., Kam, M., Regli, W., Peysakhov, M., Sevy, J., Belov, N., Morizio, N., Mroczkowski, A.: Secure mobile agents on ad hoc wireless networks. In: The Fifteenth Innovative Applications of Artificial Intelligence Conference, American Association for Artificial Intelligence (2003)
2. Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K.: MASON: A new multi-agent simulation toolkit. In: Proceedings of the 2004 SwarmFest Workshop. (2004)
3. UCB/USC/LBNL/VINT: Network simulator (NS) version 2. http://www.isi.edu/nsnam/ns/ (2003)
4. Lentini, R.P., Rao, G.P., Thies, J.N., Kay, J.: Emaa: An extendable mobile agent architecture. In: AAAI Workshop on Software Tools for Developing Agents. (1998)
5. BBN Technologies: Cougaar architecture document. http://docs.cougaar.org (2003)
6. Zimmerman, H.: OSI reference model - the ISO model of architecture for open system interconnection. IEEE Transactions on Communications **28** (1980) 425–432
7. Tennenhouse, D.L., Smith, J.M., Sincoskie, W.D., Wetherall, D.J., Minden, G.J.: A survey of active network research. IEEE Communications Magazine **35** (1997) 80–86
8. Barrett, C.L., Marathe, M.V., Engelhart, D.C., Sivasubramaniam, A.: Approximate connectivity graph generation in mobile ad hoc radio networks. In: Proceedings of the 36th Annual Simulation Symposium, IEEE Computer Society (2003) 81
9. Camp, T., Boleng, J., Davies, V.: A survey of mobility models for ad hoc network research. Wireless Communication & Mobile Computing (WCMC): Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications **2** (2002) 483–502
10. Peysakhov, M., Regli, W.: Ant inspired server population management in a service based computing environment. In: Proc. of the IEEE Swarm Intelligence Symposium. (2005)

11. Peysakhov, M., Cicirello, V.A., Regli, W.C.: Ecology based decentralized agent management system. In: Proceedings of Formal Approaches to Agent-Based Systems III. (2004)
12. Peysakhov, M.D., Lass, R.N., Regli, W.C.: Stability and control of agent ecosystems. In: Proceedings of the Fourth International Joint Conference on Autonomous Agents & Multi-Agent Systems. (2005)
13. Kopena, J.B., Naik, G., Peysakhov, M.D., Sultanik, E.A., Regli, W.C.: Service-based computing for agents on disruption and and delay prone networks. In: Proceedings of the Fourth International Joint Conference on Autonomous Agents & Multi-Agent Systems. (2005)
14. Migas, N., Buchanan, W.J., McArtney, K.A.: Mobile agents for routing, topology discovery, and automatic network reconfiguration in ad-hoc networks. In: Proceedings of the 10th IEEE Conference and Workshop on the Engineering of Computer-Based Systems. (2003) 200–206

Visual Modeling for Complex Agent-Based Simulation Systems

Candelaria Sansores, Juan Pavón, and Jorge Gómez-Sanz*

Universidad Complutense Madrid, Dep. Sistemas Informáticos y Programación,
28040 Madrid, Spain
csansores@fdi.ucm.es,{jpavon, jjgomez}@sip.ucm.es

Abstract. Currently there is a diversity of tools for agent-based simulation, which can be applied to the understanding of social phenomena. Describing this kind of phenomena with a visual language can facilitate the use of these tools by users who are not necessarily experts in computer programming, but in social sciences. With this purpose, we propose to define such visual language, which is based on well established concepts of agent-oriented software engineering, and more concretely on the INGENIAS methodology. The proposed language is independent of any particular simulation platform and, by using INGENIAS code generation support, it is possible to generate implementations for the desired target platforms. Also, we consider that modeling should be application domain oriented and that a generic language itself does not suffice. Thus, we discuss at the end how specific domain simulation environments could be achieved.

1 Motivation

Agent-based modeling and simulation is being increasingly used for exploring the complexity of social dynamics. This approach facilitates the study of how social phenomena emerge, that is, how the interactions and varied behaviors of individual agents produce structures and patterns. Thus, agent based modeling is well suited for studying systems that are composed of interacting agents and exhibit properties arising from the interactions of the agents that cannot be deduced simply by aggregating the properties of the agents. This is the kind of problems that appear when studying complex adaptive systems like those found in social sciences.

Complex adaptive systems are often characterized by agents interacting or capable of interacting with each other in dynamic, nonlinear and unpredictable ways [12]. From a computer science view, a complex adaptive system is a form of *complex* multi-agent system (MAS) with adaptive agents. However, complex MAS are rarely implemented because the system can become easily intractable. Thus, the general approach to study a complex adaptive system is a computer simulation, in which repeated iteration of *simple* local rules in a population of *simple* interacting agents leads to complex global phenomena.

* This work has been developed with support of the Consejo Nacional de Ciencia y Tecnología (CONACYT) from México and the project TIC2002-04516-C03-03, funded by the Spanish Council for Science and Technology.

J.S. Sichman and L. Antunes (Eds.): MABS 2005, LNAI 3891, pp. 174 – 189, 2006.

This drives to the fact that most agent-based models found in social science simulations are basically cellular automata, a special kind of MAS with primitive agents, arranged on a rigid grid, and interacting with one another following simple rules. Even so, emergent phenomena, like larger and perhaps global social structures and patterns of behavior, can be still observed and studied from cellular automata simulations, as many results that have been published in the social science literature show. Due to this *simplicity* in agents' implementations, some authors (for instance [8]) question whether these are really computational agents, as those found on MAS or Distributed Artificial Intelligence, and arguments sustaining agents are not used to implement agent-based simulations, but only to design them.

We agree that agents employed in most social simulations are not sophisticated agents with cognitive internal structures and communication language capabilities like the ones in MAS discipline. But we also think that these agents are, at least, minimalist agents like *tropistic agents*, whose actions are entirely determined by their current environment, or *hysteretic agents*, which retain and are influenced by their memory on previous environments and results, as well as by their current ecology [9].

For this reason, most of the agent-based social simulation tools that are available for social scientists are toolkits without a sophisticated underlying MAS model. The main goal of these tools is to assist modelers to implement simple agents that react through changes made to the environment by other agents, interacting indirectly through the environment. After a survey of simulation tools in the social science, we found that toolkits follow this philosophy, modeling simple, ecologically defined societies. Although some of these toolkits offer much more than this, not like already implemented features but providing an open implementation to integrate them.

Simplicity is not a problem if we consider modeling is a term for simplification, and the rule of Occam's razor[1] says that *the simpler the better*. However, regarding MAS discipline we question ourselves if MAS potential is not being leveraged, and the plausibility issue with regard to social simulation also arises, which, in principle, involves modeling human cognition a little more realistically.

Should agents be simple or should they be complex is a core question in the agent based modeling domain, and still an open debate. Proponents of the simplicity of agents, the so-called *keep-it-simple-and-stupid*, or KISS principle [3], point out that the most interesting analytical results are obtained when complexity at the macro level is produced by simple micro-level dynamics.

Proponents of the complexity of agents [5] obtain their arguments especially from the fields of Sociology and Cognitive Psychology, and emphasize the idea that agents should be kept as simple as suitable. There are uses of computer simulation in which the faithful reproduction of a particular setting is important, like prediction of interest rates in economy. For this purpose, the assumptions that go into the model may need to be quite realistic.

In this work we do not pretend to provide an answer to the simplicity vs. complexity discussion, neither to provide another simulation toolkit; we consider that

---

[1] Occam's razor is a logical principle (often called the *principle of parsimony*) attributed to the mediaeval philosopher William of Occam. The principle states that one should not make more assumptions than the minimum needed. It admonishes us to choose from a set of otherwise equivalent models of a given phenomenon the simplest one.

there are good tools already, each one providing different features and bringing a unique perspective to the field. This diversity gives many benefits in terms of affording the opportunity to explore new ideas within the various toolkits and providing various approaches that fit with different types of problems.

Our purpose is to facilitate the use of these tools by social scientists, and the way to do this is to abstract them, as much as possible, from simulation programming issues. According to this, we intend to provide an easily customizable tool for describing phenomena in different social domains with a visual language. This visual language must be toolkit-independent for specifying simulation models. It supports simple agents (like those already supported by existing agent-based simulation toolkits) and complex agents (that mimic real agents as much as necessary), and considers organizational issues to manage another dimension of complexity (the society of agents and the system's architecture).

Initially, we have identified some open issues in social sciences simulation literature and on several surveys we have carried out in the field as well as others like [13, 23]. The need for a toolkit-independent visual modeling language is mainly supported by model communication and replication requirements. A common language to describe agent-based models, independent of the implementation platform, could be a better mean of communication among users or tools than using source code, since it demands a high effort to understand and interpret the models, sometimes leading to misunderstandings.

Also, we are aware of having a universal modeling and simulation language for the social domain is not feasible, as it has been already discussed in [10], so our aim is to provide a language that can be customized to particular social domains, by specialization or addition of new elements, which can be defined by the modelers of such domains. With this purpose we adopt an agent-oriented methodology, INGENIAS [16]. This methodology is based on meta-modeling principles and supported by a set of tools for modeling and code generation for multiple platforms. In this sense, it can facilitate our approach as it can be easily adapted for our purposes by extending its underlying meta-model.

Section 2 reviews approaches for agent based simulation modeling. Section 3 briefly describes the INGENIAS agent oriented methodology, which is adapted to model agent-based social systems. Section 4 presents a case study that shows the feasibility of describing complex agent-based simulation systems with the INGENIAS visual modeling language. Section 5 summarizes our contributions and concludes with the identification of further work.

2 Agent Based Simulation Modeling

As agent-based simulation has gained in popularity, software tools for modelers are emerging. There are various toolkits for developing agent-based simulation systems. REPAST [17] is a set of Java libraries that allows programmers to build simulation environments, to create agents in social networks, to collect data from simulations automatically, and to build user interfaces easily. Its features and design owe a lot to SWARM [22], one of the first agent-based modeling libraries. Another similar library is ASCAPE [2]. MASON [14] is a single-process discrete-event simulation core and visualization toolkit written in Java, designed to be flexible enough to be used for a wide range of simulations, but with a special emphasis on swarm simulations of a huge number (up to millions of) agents. The design philosophy of these toolkits is to provide a model library to which an experienced programmer can easily add features for simple simulations.

These libraries have great advantages for modelers over developing their own, but they require having a good working knowledge of a programming language (usually Java). Thus, inexperienced programmers could find that a higher-level declarative language instead of a programming language would be desirable. This was the purpose of SDML [19], which has been built on Smalltalk. Unlike ASCAPE and REPAST, it does not demand users to be fluent in the underlying programming language, but they have to learn a complex interface that can be as difficult to master as a full programming language, which finally limits its usability.

Another kind of tool that has emerged for developing simulations is rapid development environments. These allow the building of simple models using visual programming, for instance, STARLOGO [21], NETLOGO [15], CORMAS [6] and AGENTSHEETS [1]. Although they are relatively easy to use, agents in this kind of systems are quite simple, usually with a poor or nonexistent agents' cognitive model, and without support to model direct interactions between agents.

Simulations developed with both kinds of tools, either toolkits or rapid-development libraries, are proprietary model descriptions. This makes impracticable to compare two different implementations of the same specification. Therefore, replication of models turns out complicated due to the diversity of specification languages and the lack of a common one that addresses conceptual modeling. One approach that seems to consider this requirement more formally is SESAM [20]. This framework provides an environment for modeling, which is based on UML-like activity diagrams. Although it does a step forward in the modeling issue, it still lacks of common functionality that is usually available in more general agent development tools, such as the ability to specify agents direct interactions via messages. Thus, interactions in such simulations are expressed indirectly through changes in the environment, mainly because this toolkit is based on reactive agents and not on sophisticated internal reasoning process, which are prerequisites for complex negotiation abilities. So, it does not support complex goal-oriented agents.

Our purpose is to increase the potential of agent-based simulation tools by providing a visual language for describing agent-based simulation models and abstracting the model specification from the implementation platform.

3 A Visual Modeling Language Based on INGENIAS

INGENIAS [16] provides a language for modeling multi-agent systems (MAS), and support tools for analysis, design, verification and code generation, the INGENIAS Development Kit (IDK). These tools, as well as the language, are based on the specification of meta-models that define the different views and concepts from which a multi-agent system can be described. By extending and refining the meta-models it is possible to adapt the language and the tools for particular application domains, always relying on agent concepts. As a proof of concept, this has been applied, for instance, for holonic manufacturing systems [11]. This is one of the main reasons to consider INGENIAS as the basis for our approach.

A MAS specification with INGENIAS is performed from five viewpoints, whose use is illustrated in the next section. An important characteristic of INGENIAS is that, when describing a MAS, it fully assumes the social and intentional nature of agents.

- **Organization viewpoint.** The organization describes the framework where agents, resources, tasks and goals coexist. It is defined by its structure, functionality and social relationships. From a structural viewpoint, the organization is a set of entities with relationships of aggregation and inheritance. The organization structure defines a decomposition of the MAS in *groups* and *workflows.* Workflows define the dynamics of the organization, by describing consumer/ producer associations between tasks as well as assignment of responsibilities (to agents or roles) for their execution, and resources associated to each one. Social relationships define dependencies (for instance, subordination, peer-to-peer) between agents in the organization. Thus, the organization allows managing complexity by structuring the system and determining global behavioral rules.
- **Agent viewpoint.** Each agent is defined by its purpose (what goals an agent is committed to pursue), responsibilities (what tasks it has to execute), and capabilities (what roles can play). The way this is reflected is by defining the mental state of the agent, its management and processing. The mental state consists of goals of the agent, the information it has about the satisfaction of those goals, and its knowledge about the world and facts reflecting its past experience. Thus, the mental state provides the information that the agent needs to make decisions, i.e., to decide what actions to execute.
- **Environment viewpoint.** The environment is defined in terms of what agents can perceive or actuate. It also identifies available resources as well as already existing agents and applications with which an agent can interact.
- **Tasks and Goals viewpoint.** It considers the break down of goals and tasks. It also describes the consequences of performing a task, and why it should be performed (i.e., it justifies the execution of tasks as a way to satisfy goals). For each task, it determines what elements are required and what outputs are expected. To identify which goals are influenced by a task execution, there are satisfaction and failure relationships. Finally, the tasks/goals viewpoint explains how a solved goal affects other existing goals by using decomposition and dependency relationships.
- **Interaction viewpoint.** The interaction viewpoint addresses the exchange of information or requests between agents, or between agents and human users. The definition of an interaction in INGENIAS goes a step further than traditional formalisms, such as UML sequence diagrams, in the sense that it reflects the motivation of the interaction and its participants. It also includes information about the mental state required by each agent throughout the interaction as well as tasks executed in the process. In this way, it allows to express at design level why an agent engages in an interaction and why it should continue. Furthermore, interaction protocols can be specified using different formalisms: AUML, UML collaboration diagrams and *GRASIA* diagrams, a specialization of UML collaboration diagrams to address intentional issues associated with an interaction.

It is important to notice that INGENIAS is an open methodology for modeling MAS, and these viewpoints just provide a guideline to build models during the analysis and design phase; this implies that the developer has the responsibility to decide which models generate, in which order and the level of detail. In this sense, our purpose here is to see whether INGENIAS provides enough modeling elements for complex agent-based simulation systems, which issues require some refinement or extensions in order to specify time properties and location of agents in the environment, and deployment of experimental models. This has driven us to consider two new viewpoints: the Timer viewpoint and the Space viewpoint, which can be considered as extensions of the Environment viewpoint.

The Timer viewpoint manages the flow of time in the model during the execution of the simulation. Because our approach assumes simulations that are going to be carried out are time driven, we need constant time steps for simulating the perception-reaction cycle of agents who act by the passage of time. The Space viewpoint describes the spatial arrangements of agents in the model. Currently, we have only defined simple spatial relationships like continuous spaces, but we plan to consider 2D, 3D, hexagonal grids, hexagonal tori, etc., spaces, as in most simulation toolkits.

The case study in the next section shows in practice a set of models and design artifacts with this modeling language. We will see the details of each model considering that they are not only blueprints but development artifacts that contribute directly to the implementation phase of the simulation system.

4 The Case Study

The example used to illustrate the visual modeling language for agent-based simulation systems is taken from [7], which studies altruism among simple and smart vampire bats. This example is interesting as it shows the importance of modeling agents as cognitive entities and remarks the impact of intelligence, goal-based systems on the spreading of altruism, provided these systems are highly dynamic. The model considers a population of vampire bats (agents) that live in *roosts*, where they get back to after hunting and perform social activities like grooming and sharing food. The entities explicitly modeled as agents are the bats. Roosts are modeled as aggregates of bats. In roosts, bats are allowed to share food and to groom one another.

Each simulation cycle includes one daily and one nightly stage. During the daily stage, the simulated animals perform the social activities, in the night they hunt. Each night 93% of the population will find food to survive until the next hunt. The remaining 7% will starve, unless they receive help from some fellow (under the form of regurgitation). A starving bat will turn to grooming partners for help, and will avoid death if any of them is found to be full. Help-giving allows animals to achieve credits, which will be extinguished if and when help is returned. Animals are endowed with social knowledge: consisting of a memory of past grooming and food-sharing interactions, and of consequent credits. After each simulation the number of death bats, the number of altruistic acts performed and the number of credits turned on or off are recorded.

We start modeling the case study with an Organization model. The organization allows managing complexity by structuring the system and determining global behavioral rules. At a global level, the goal of the organization (its purpose) is shared by its agents, which can be structured in groups. Agents in the organization execute tasks that are identified in workflows or distributed plans, which depict the dynamics of the organization.

In the case study, the organization is a *bats population*, whose purpose is the goal *Survive*, as shown in Fig. 1. This organization consists of one or more *roosts* (represented as groups with the group icon) and a *credit network* (represented as an organization network with the network icon). There is only one type of agents, Bats, which can play several roles. Bats belong to *roost* group. In a roost, a bat can play the role of *altruist* or *recipient* depending on whether help is given or received. In the credit network, a bat can play the role of *NodeTo* and *NodeFrom*.

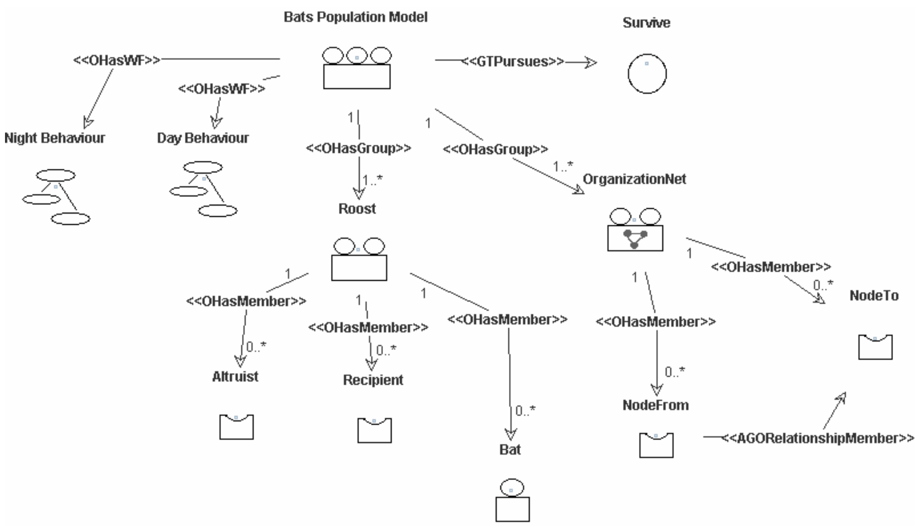


Fig. 1. The organizational model diagram. The organization (represented as a rectangle with three circles) pursues a goal (represented as a circle), has groups (represented as rectangles with two circles) and workflows or plans. Agents (a rectangle with one circle) and roles (similar to agents but without a circle) are members of one or more groups.

The cardinality in the associations indicates the organization can have one or more roosts and the roost one or more bats, the same applies to roles cardinality, which indicates that it is possible that the role of recipient, for instance, may not be played by any agent.

The behavior of bats is represented collectively as *plans*, *Night Behavior* and *Day Behavior* in Fig. 1, which are further broken down in tasks.

The *Organization Network* entity (represented as rectangles with two circles and three nodes inside in Fig. 1) is a clear example of how the INGENIAS modeling language can be extended. We added this new entity to support network models, since these models are frequently studied in social simulation systems.

The entities which have been first identified in the organizational model can be further detailed when looking at the system from other viewpoints, in other diagrams, as it is shown below.

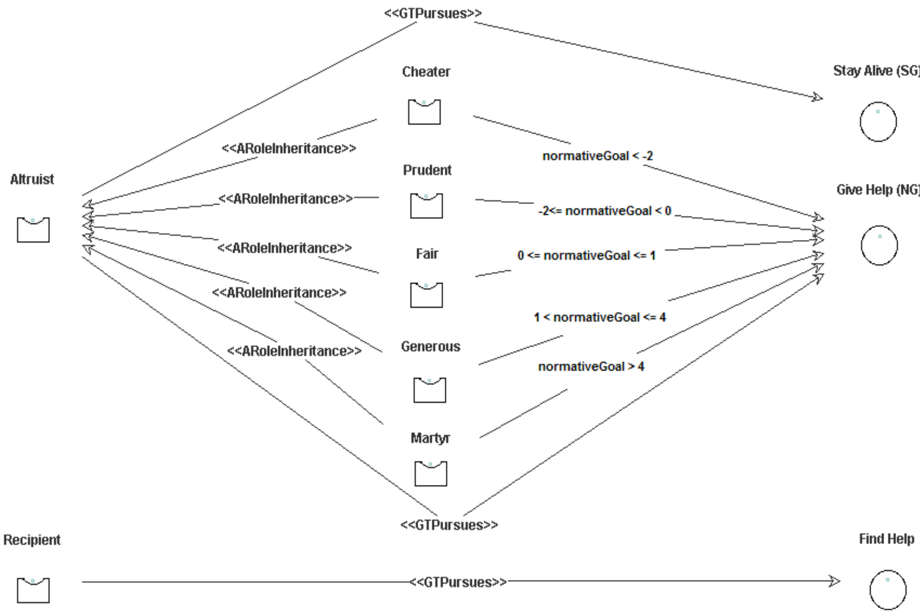


Fig. 2. The roles/goals model diagram. Depending of bats' motivational force for pursuing goals each agent plays a different role. This is a function of the motivational value for pursuing SG and NG goals and is a good example of roles specialization and inheritance.

The simulation model we consider in this case study is the dynamic variant of smart agents in [7], this states that smart agents are endowed not only with actions, but also with mental states such as beliefs and goals. Therefore, the norm of reciprocity is a more complex construct taking beliefs as input conditions and giving a set of mental representations, namely beliefs and goals, as outputs. The main output of the rule is then a normative belief; it tells what is expected of oneself. Whether we will actually conform or not depends upon a further mental process initialized by the rule, this is, the formation of a *normative goal* based upon the *normative belief* [4].

Therefore, in this model, goal-dynamics is an essential aspect of cognition. Agents are endowed with a cognitive architecture in which a goal is a mental construct. More specifically, in this case, agents have two goals, the normative (*give help*) and the survival goal (*stay alive*). Both goals vary according to their motivational force. The smart algorithm implementing the mental state has a small number of rules modifying the values of agents' motivational force, in particular of the normative goal (NG). The motivational force of survival goal (SG) has been fixed to 0, while the normative motivation is left free to change. The normative value is incremented by altruistic acts: the experience of an effective application of the norm (donation and, to a greater amount, reciprocation) raises the value of normative motivation, while the increase of unreciprocated donations lowers it. As a consequence, the strategy applied by an agent is dictated by the value of its *normative motivation*.

These issues are described in the diagram in Fig. 2, a roles/goals model. This model is important because it shows role dynamics based on motivational force for pursuing goals. If motivational force for pursuing certain goal changes, then an agent changes dynamically the role is playing (through roles inheritance).

As shown in Fig. 2, a bat agent can play two different roles, each one defined pursuing different goals. In this case, there is a major role of *altruist* which pursues goals *Stay Alive* and *Give Help*. Depending of motivational force of giving help expressed with the fact *normativeGoal*, bats can play an inherited role of the altruist: *cheater* when *normativeGoal* <-2, *prudent* when -2<= *normativeGoal* <0, *fair* when 0<= *normativeGoal* <=1, *generous* when 1< *normativeGoal* <=4, and *martyr* when *normativeGoal* >4.

Also, there is a second role of *recipient*, this role is played by starving bats that ask for help and are given blood to survive; this role pursues the goal *Find Help*. The *NodeTo* and *NodeFrom* roles are special roles played by the agent although they are not shown in this model; however, they are implicit in the implementation and it is not necessary to include them in the diagram.

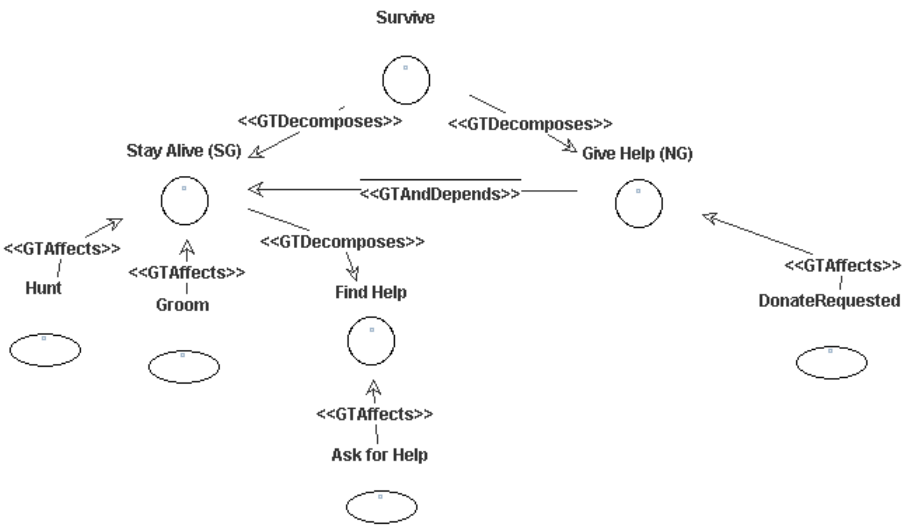


Fig. 3. The tasks/goals model diagram. This model shows the motivations behind tasks execution and how they affect goals.

Goals should be broken down into sub goals up to a level where concrete tasks can be specified. This is modeled in the tasks/goals model, which justifies the execution of tasks as a way to satisfy goals. The association of objectives and tasks is guided by the *principle of rationality,* in which agent actions are justified by the objectives it pursues.

Also, tasks in this model are important because they play an important role in the evolution of mental states of its executors. In a more pragmatic point of view, tasks are a set of instructions to be executed and could be integrated in a plan. Thus, tasks shown in this model belong to workflows or plans depicted in the organizational model diagram.

Fig. 3 shows the relationship between goals and tasks. Here, we can observe how the S*urvive* goal breaks down into two sub-goals: *Stay Alive* goal and *Give Help* goal which depends on the former. Tasks are associated with objectives through *GTAffects* relationship, whose semantics indicates that a task affects the mental state of its executors, just like we said before. In the same way, S*tay Alive* goal decomposes in *Find Help* goal which is affected by task *Ask for Help*.

Now, we define each type of agent in the organization with an agent model. In this model each agent is defined by its responsibilities (goals is committed to pursue, tasks it has to execute, and roles can play) and behavior. Agent's behavior is specified by defining the mental state of the agent, as well as its mental state management and processing. As we said before, the mental state entities consist of: agents' goals and the information it has about the satisfaction of those goals, its knowledge about the world and facts reflecting its past experience.

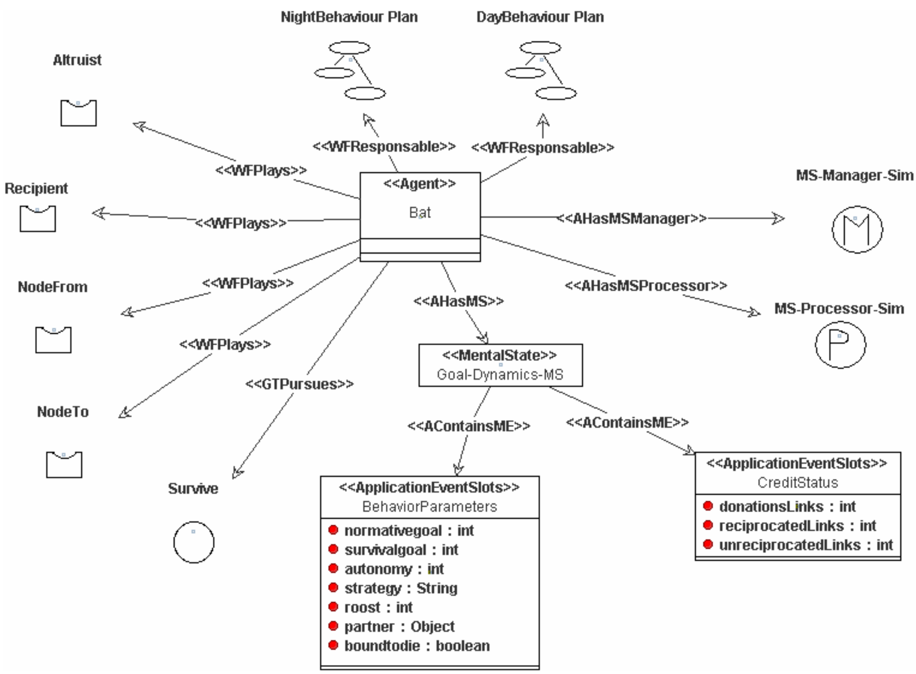


Fig. 4. The agent model diagram. Depict agent's mental state entities, like goals, roles, tasks, and facts, as well as its mental state manager and processor.

The mental state is managed and processed by a mental state manager and processor to produce agents' decisions. The manager provides for operations to create, destroy and modify mental state entities, and the processor take decisions based on goals and tasks that satisfy them. In Fig. 4 both entities are represented with the *MS-Manager-Sim* and *MS-Processor-Sim* icons respectively.

The separation between agent's responsibilities and behavior allows to use different mechanisms like neural networks, learning algorithms, and user defined ones, etc. to implement agent's mental state control mechanisms. In this work, we defined our own appropriated mental state manager and processor which are represented in Fig. 4.

In this way, the final user does not need to define these complex elements, but only the mental state entities.

Also, we can see in Fig. 4 the agent's mental state informational entities named *Goal-Dynamics-MS* with facts represented in frames (*ApplicationEventSlots*), as well as goals pursued by the agent, in this case the *Survive* goal, roles played by the agent: *altruist*, *recipient*, *NodeFrom*, *NodeTo*, and plans performed by the agent, like *Night-Behaviour Plan* and *DayBehaviour Plan*.

Tasks play a fundamental role in the evolution of agents' mental state, since It describes how the mental state of agents change over time, what is the consequence of executing a task with respect the mental state of an agent, how to achieve goals and what happens when a goal cannot be achieved. Therefore we define a tasks model to represent this information.

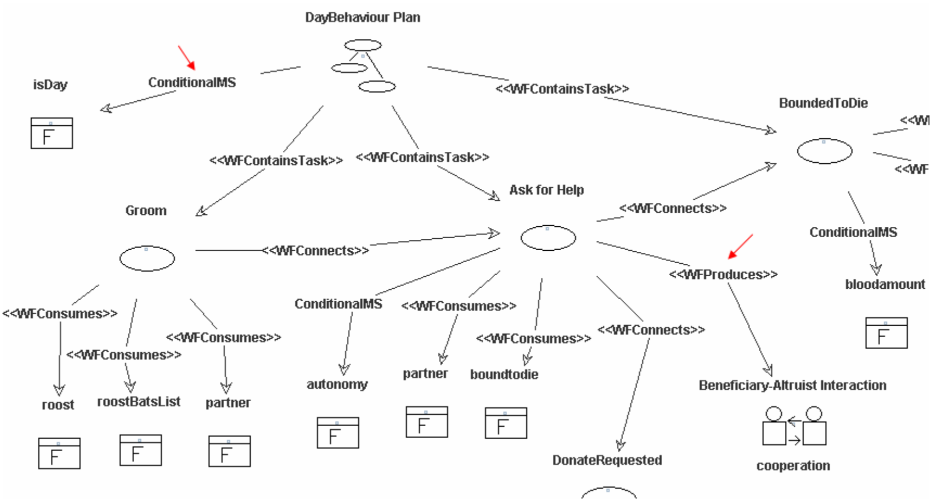


Fig. 5. An extract of the tasks model diagram. Represent a plan of actions an agent can execute, the plan is decomposed in individual tasks which can consume or produce certain facts. The execution of a task can produce interactions as well.

The extract of the tasks model in Fig. 5 defines a plan of actions executed by the agent. The plans are expressed like workflows with a sequence of tasks, here the precedence of tasks and the condition required to be executed are established by relationships like *ConditionalMS*, *WFConnects*, *WFDecomposes*, etc.

The conditional relationship indicates there is a specific mental state required by the agent to be able to execute the plan or task, this condition is further detailed in a conditional mental state model diagram. Also, tasks can produce interactions, like we can see in the figure and which are further described in an interaction model diagram.

In Fig. 5, the execution of *DayBehaviour Plan* is conditioned by fact *isDay* having certain value, and the condition is described in a conditional mental state model diagram. The plan contains tasks *Groom*, *Ask for Help*, and *BoundedToDie*. Task *Ask for Help* consumes several facts, connects with other tasks and produces interaction *Beneficiary-Altruist Interaction*, as can be seen in Fig 5.

A conditional mental state model is actually an instance of an agent in runtime and is also known as instantiated agent model. An instance of an agent in runtime is necessary to describe the evolution of the agent's mental state referring to properties of running agents, like restrictions or requirements an agent should satisfy. Also it is useful for denoting a concrete agent in the system.

At the bottom of Fig. 6 we can see a conditional mental state model diagram which depicts the condition to be met in order to execute *DayBehaviour Plan*. The model consist of an agent type between brackets which represents an instance of an agent in runtime, this instance has the fact *isDay* which participate in the description of requirements or conditions for its mental state. We used a Java expression to describe the condition, which says: *isDay* fact should be true.

This conditional mental state model is associated to *DayBehaviour Plan* editing *ConditionalMS* relationship in the task model and adding the name of the conditional mental state model in the editing box, as shown at the top of Fig. 6.

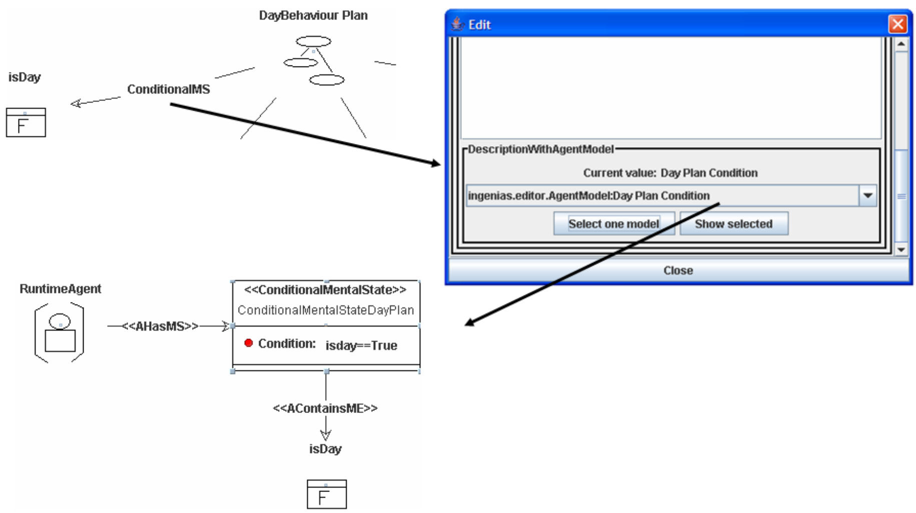


Fig. 6. The conditional mental state model diagram. At the bottom of the figure we can observe the conditional mental state model with a description of requirements for an instance of an agent's mental state in runtime. The relationship of this model and the task model is shown at the top of the figure.

A traditional system consists of several conditional mental state models and used to be more complex, including for example roles the agent should be playing or goals should be pursuing in order to meet the condition.

Another aspect modeling simulation systems is the specification of agent's interactions. An interaction is defined by two models, a contextual model, as the one we can see in Fig. 7, and a model with the interaction specification details, that is, the interaction model, as the one in Fig 8.

Contextual information specifies what goals are pursued by the interaction, the actors that participate, their motivations to engage in the interaction and constraints on

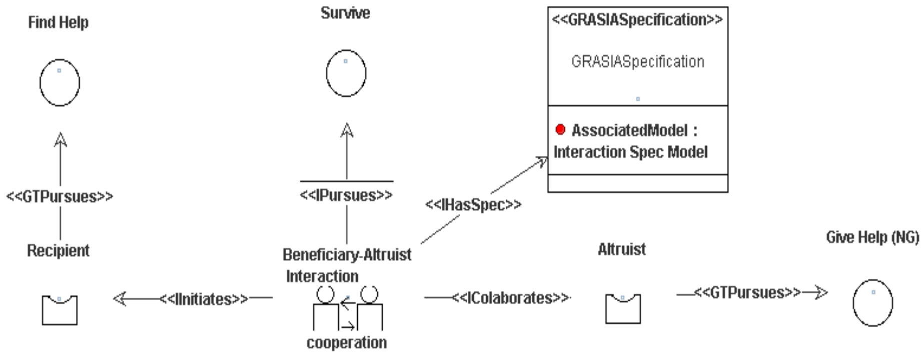


Fig. 7. The contextual model diagram. This model specifies contextual information of interactions.

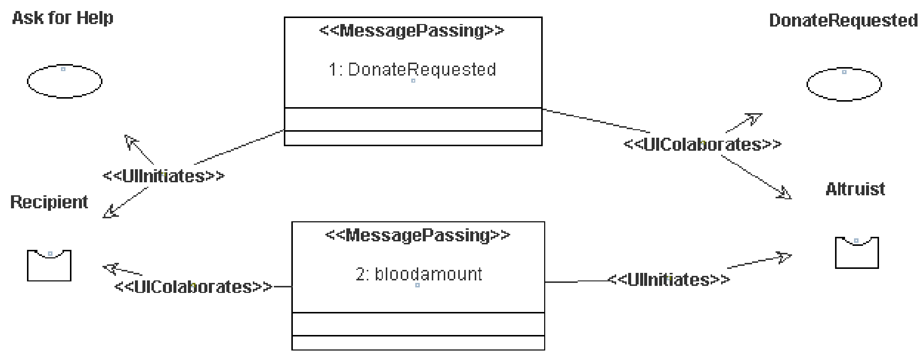


Fig. 8. The interaction model diagram. Consist of two participants, one of them initiates the interaction and the other one collaborates, the model show the sequence of messages and the tasks participants perform during the interaction.

their mental states. In Fig. 7 we can see the *Recipient* role pursuing goal *Find Help* and initiating an interaction with a collaborative agent playing role of A*ltruist* and pursuing goal *Give Help*. The interaction goal is the *survival* of participants. The interaction details are specified with a *GRASIA* specification diagram (see Sec. 3) and is indicated with a *HasSpec* relationship as is shown in Fig. 7 and the name of the interaction model in the associated model attribute.

The interaction model in Fig. 8 consists of the exchange of interaction units in a sequence (e.g., messages) between a starving bat and a potential altruist. When a bat is starving, it plays role of *Recipient* and executes *Ask for Help* task which produces an interaction, that is, initiate an interaction. *DonateRequested* message is passed to *Altruist* role which collaborates in the interaction and executes task *DonateRequested* to decide if it can help and the amount of blood to give away, sending back a message with *bloodamount* to *Recipient*.

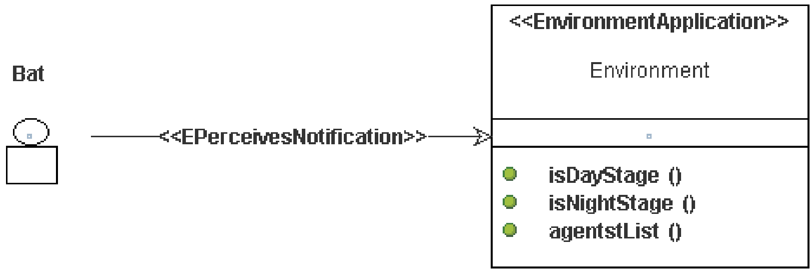


Fig. 9. The environment model diagram. Depict a bat agent perceiving changes in the environment as well as already existing agents on it through configurable attributes.

Finally, the environment model is defined. Fig. 9 shows what a bat agent can perceive through changes in the environment, like current stage, either day stage or night stage. It also identifies already existing agents. Information perceived by agents is configured with different attributes in the *Environment* entity, in this case: *agentList*, *isDayStage* and *isNightStage* attributes. Time and Space attributes are not shown in this model but they are implicit like implementation issues and are configurable during deployment phase.

With this set of models it is possible to specify a simulation system with a view of the global structure (organization), the different roles and goals pursued by system entities, and the behavior of the entities, both individually (in terms of the management and processing of their mental state) and collectively (in terms of interactions between agents and with the environment). The next step is code generation from the models. The code generation process is based on templates, which should be previously developed for the target simulation toolkit. These templates are instantiated taking information from the MAS model. In concrete, agent mental state manager and processor are the most variable part in different mappings for specific target platforms. Interactions can be modeled through the implementation of specific mechanisms in the target platform. These techniques have been already used in the INGENIAS Development Kit for several target platforms such as JADE, Robocode or J2EE, therefore we are confident about its feasibility.

5 Conclusions

In this paper we consider the need for the visual modeling of social systems composed of multiple agents with complex behaviors, which can be based on intentional and organizational concepts. The specification language is more declarative than procedural, which in principle should facilitate model communication between different users. Given that this kind of modeling language is independent of specific simulation toolkit or library, it should also facilitate model replication.

We have considered a direct application of INGENIAS language, just to check that agent oriented software engineering offers enough elements to specify the kind of problems that social simulation requires. However, we are conscious that this kind of language, although visual, is not usable enough for non experts in the agent approach. Therefore, we envision developing from these concepts, application domain oriented versions, in which those agent concepts are wrapped and represented in terms of application domain concepts. This is in line with INGENIAS approach, which is based on the application of meta-models, whose manipulation allows developing domain-specific tools.

The second issue is the generation of code for specific target platforms, which is also related with the replication problem. Our approach is to use the INGENIAS Development Kit (IDK) for building code generation modules. Basically, once basic templates for specific target platform are defined, the IDK offers an API and libraries to navigate the specification graph in order to determine how to instantiate specification elements with the templates. At the end, the modeler will have a customized visual environment to specify simulation models that will be executed on one or several simulation engines, with the ability to contrast results.

After the visual modeling phase that has been presented here with a case study, the implementation on multiple platforms for REPAST and MASON toolkits is discussed in [18]. We have started with these two toolkits because in the case of REPAST it has demonstrated to be a suitable simulation framework widely used in different MABS domains, and MASON, although recently appeared, it has adequate features to create complex simulations with a huge amount of agents. Besides, both tools are open Java libraries, which is very suitable for the INGENIAS code generation approach.

References

[1] AgentSheets. 2004. AgentSheets: Agent-based end-user development. Available from: http://www.agentsheets.com

[2] Ascape. 2000. Ascape: Agent based research. Available from: http://www.brook.edu/es/dynamics/models/ascape

[3] Axelrod, R. *Advancing the Art of Simulation in the Social Sciences*. In: R. Conte, et al. (Eds.): *Simulating Social Phenomena*. Lecture Notes in Economics and Mathematical Systems, Vol. 456. Berlin Springer. (1997), 21-40.

[4] Conte, R., C. Castelfranchi. *Cognitive and Social Action*. 1995, London: UCL Press.

[5] Conte, R. *Social Intelligence Among Autonomous Agents.* Computational and Mathematical Organization Theory, 1999. Vol. 5(3): p. 203.

[6] Cormas. 2004. Cormas: Natural resources and agent-based simulations. Available from: http://cormas.cirad.fr/indexeng.htm

[7] Di Tosto, G., R. Conte, M. Paolucci. *Altruism Among Simple and Smart Vampires*. In: *Proceedings of the 1st Conference of the European Social Simulation Association (ESSA)*. Groningen, NL (2003),

[8] Drogoul, A., D. Vanbergue, T. Meurisse. *Multi-agent Based Simulation: Where Are the Agents?* In: J.S. Sichman, et al. (Eds.): *Multi-Agent-Based Simulation II: Third International Workshop, MABS 2002*. Lecture Notes in Computer Science, Vol. 2581. Springer. Bologna, Italy (2002), 1-15.

[9] Ferber, J. *Multi-agent systems: an introduction to distributed artificial intelligence*. 1999, Harlow: Addison-Wesley.

[10] Gilbert, N. *Environments and Languages to Support Social Simulation*. In: K.G. Troitzsch, et al. (Eds.): *Social Science Microsimulation*. Berlin Springer. Schloss Dagstuhl (1996), 457-458.

[11] Giret, A., V. Botti, S. Valero. *MAS Methodology for HMS*. In: *Second International Conference on Applications of Holonic and Multi-Agent Systems, HoloMAS 2005*. 3593. Springer-Verlag. (2005), 39-49.

[12] Holland, J.H. *Emergence: from chaos to order*. 1998, Reading, Massachusetts: Addison-Wesley.

[13] Marietto, M.B., N. David, J.S. Sichman, H. Coelho. *Requirements Analysis of Agent-Based Simulation Platforms: State of the Art and New Prospects*. In: J. Sichman, et al. (Eds.): *Multi-Agent-Based Simulation II: Third International Workshop, MABS 2002*. Lecture Notes in Computer Science, Vol. 2581. Springer. Bologna, Italy (2002), 125-141.

[14] Mason. 2004. Mason: Multiagent Simulation Toolkit. Available from: http://cs.gmu.edu/~eclab/projects/mason/

[15] NetLogo. 2004. NetLogo: Center for Connected Learning and Computer-Based Modeling. Available from: http://ccl.northwestern.edu/netlogo/

[16] Pavón, J., J. Gómez-Sanz. *Agent Oriented Software Engineering with INGENIAS*. In: V. Marík, et al. (Eds.): *Multi-Agent Systems and Applications III, 3rd International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS 2003*. Lecture Notes in Computer Science, Vol. 2691. Springer. Prague, Czech Republic (2003), 394-403.

[17] Repast. 2004. Repast: Recursive Porus Agent Simulation Toolkit. Available from: http://repast.sourceforge.net

[18] Sansores, C., J. Pavón. *Agent Based Simulation for Social Systems: from Modeling to Implementation*. In: *11th Conference of the Spanish Association for Artificial Intelligence*. Lecture Notes in Computer Science, Vol. Springer-Verlag. Santiago de Compostela, Spain (2005), To appear.

[19] SDML. 1997. SDML: a Strictly Declarative Modelling Language. Available from: http://www.cpm.mmu.ac.uk/sdml

[20] SeSAm. 2004. SeSAm: Integrated Environment for Multi-Agent Simulation. Available from: http://www.simsesam.de/

[21] StarLogo. 2004. StarLogo: on the web. Available from: http://education.mit.edu/starlogo/

[22] Swarm. 2004. Swarm Wiki: the agent-based modelling resource. Available from: http://wiki.swarm.org

[23] Tobias, R., C. Hofmann. *Evaluation of free Java-libraries for Social-Scientific Agent Based Simulation.* Journal of Artificial Societies and Social Simulation, 2004. Vol. 7(1).

Author Index