LNAI 7879

Yves Demazeau Toru Ishida
Juan M. Corchado Javier Bajo (Eds.)

Advances on Practical Applications of Agents and Multi-Agent Systems

11th International Conference, PAAMS 2013
Salamanca, Spain, May 2013
Proceedings

Springer

Yves Demazeau Toru Ishida
Juan M. Corchado Javier Bajo (Eds.)

Advances on Practical Applications of Agents and Multi-Agent Systems

11th International Conference, PAAMS 2013
Salamanca, Spain, May 22-24, 2013
Proceedings

Springer

Series Editors

Randy Goebel, University of Alberta, Edmonton, Canada
Jörg Siekmann, University of Saarland, Saarbrücken, Germany
Wolfgang Wahlster, DFKI and University of Saarland, Saarbrücken, Germany

Volume Editors

Yves Demazeau
Centre National de la Recherche Scientifique
Grenoble, France
E-mail: yves.demazeau@imag.fr

Toru Ishida
Kyoto University, Japan
E-mail: ishida@i.kyoto-u.ac.jp

Juan M. Corchado
Universidad de Salamanca, Spain
E-mail: corchado@usal.es

Javier Bajo
Universidad Politécnica de Madrid, Spain
E-mail: javier.bajo@upm.es

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-38072-3 e-ISBN 978-3-642-38073-0
DOI 10.1007/978-3-642-38073-0
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: Applied for

CR Subject Classification (1998): I.2.11, I.2, I.6, K.3, K.4, J.1, J.2, J.7, C.2, K.4.4, H.3, H.4

LNCS Sublibrary: SL 7 – Artificial Intelligence

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Research on agents and multi-agent systems has matured during the last decade and many effective applications of this technology are now deployed. An international forum to present and discuss the latest scientific developments and their effective applications, to assess the impact of the approach, and to facilitate technology transfer has become a necessity and was in fact created a few years ago.

PAAMS, the International Conference on Practical Applications of Agents and Multi-Agent Systems, is the international yearly platform to present, to discuss, and to disseminate the latest developments and the most important outcomes related to real-world applications. It provides a unique opportunity to bring multi-disciplinary experts, academics, and practitioners together to exchange their experience in the development and deployment of agents and multi-agent systems.

This volume presents the papers that were accepted for the 2013 edition of PAAMS. These articles report on the application and validation of agent-based models, methods, and technologies in a number of key application areas, including: agents for real-world Problems; crowds modelling and analysis; decision making and discovery; interaction with artificial agents; mobility, ubiquity, and clouds; (multi-)agent design technology; and simulation and organization. Each paper submitted to PAAMS went through a stringent peer-review by three members of the international committee composed of 93 internationally renowned researchers from 24 countries. From the 70 submissions received, 14 were selected for full presentation at the conference; another nine papers were accepted as short presentations. In addition, a demonstration session featuring innovative and emergent applications of agent and multi-agent systems and technologies in real-world domains was organized. In all, 16 demonstrations were shown, and this volume contains a description of each of them.

We would like to thank all the contributing authors, the members of the Program Committee, the sponsors (IEEE SMC Spain, IBM, AEPIA, AFIA, University of Salamanca and CNRS), and the Organizing Committee for their hard and highly valuable work. Their work has helped to contribute to the success of the PAAMS 2013 event. Thanks for your help - PAAMS 2013 would not exist without your contribution.

Yves Demazeau
Toru Ishida
Juan Manuel Corchado
Javier Bajo

Organization

General Co-chairs

Yves Demazeau	Centre National de la Recherche Scientifique, France
Toru Ishida	University of Kyoto, Japan
Juan M. Corchado	University of Salamanca, Spain
Javier Bajo	Polytechnic University of Madrid, Spain

Advisory Board

Frank Dignum	Utrecht University, The Netherlands
Jörg P. Müller	Technische Universität Clausthal, Germany
Juan Pavón	Universidad Complutense de Madrid, Spain
Michal Pěchouček	Czech Technical University in Prague, Czech Republic

Program Committee

Carole Adam	University of Grenoble, France
Frederic Amblard	University of Toulouse, France
Francesco Amigoni	Politecnico di Milano, Italy
Luis Antunes	University of Lisbon, Portugal
Javier Bajo	Polytechnic University of Madrid, Spain
Jeremy Baxter	QinetiQ
Michael Berger	Docuware AG, Germany
Olivier Boissier	Ecole Nationale Superieure des Mines de Saint Etienne, France
Vicente Botti	Polytechnic University of Valencia, Spain
Lars Braubach	Universität Hamburg, Germany
France Brazier	TU Delft, The Netherlands
Stefano Bromuri	University of Applied Sciences Western Switzerland
Valerie Camps	University of Toulouse, France
Longbing Cao	University of Technology Sydney, Australia
Javier Carbo	Carlos III University of Madrid, Spain
Lawrence Cavedon	RMIT Melbourne, Australia
Pierre Chevaillier	University of Brest, France
Helder Coelho	University of Lisbon, Portugal
Juan Manuel Corchado	University of Salamanca, Spain
Keith Decker	University of Delaware, USA

Yves Demazeau	Laboratoire d'Informatique de Grenoble, France
Frank Dignum	Utrecht University, The Netherlands
Virginia Dignum	TU Delft, The Netherlands
Alexis Drogoul	Institut de Recherche pour le Développement, Vietnam
Julie Dugdale	University of Grenoble, France
Amal Elfallah	University of Paris 6, France
Maksims Fiosins	Clausthal University of Technology, Germany
Klaus Fischer	DFKI, Germany
Rubén Fuentes	Complutense University of Madrid, Spain
Sylvain Giroux	University of Sherbrooke, Canada
Marie-Pierre Gleizes	University of Toulouse, France
Pierre Glize	University of Toulouse, France
Daniela Godoy	ISISTAN, Argentina
Jorge Gomez-Sanz	Complutense University of Madrid, Spain
Vladimir Gorodetski	University of Saint Petersburg, Russia
Olivier Gutknecht	ACM, USA
Kasper Hallenborg	University of Southern Denmark, Denmark
Koen Hindriks	University of Delft, The Netherlands
Benjamin Hirsch	Technical University of Berlin, Germany
Martin Hofmann	Lockheed Martin, USA
Tom Holvoet	Catholic University of Leuven, Belgium
Shinichi Honiden	National Institute of Informatics Tokyo, Japan
Jomi Fred Hubner	Universidad Federale de Santa Catarina, Florianopolis
Toru Ishida	University of Kyoto, Japan
Takayuki Ito	Massachusetts Institute of Technology, USA
Michal Jakob	Czech Technical University in Prague, Czech Republic
Vicente Julian	Polytechnic University of Valencia, Spain
Achilles Kameas	University of Patras, Greece
Takahiro Kawamura	Toshiba, Japan
Stefan Kirn	Universität Hohenheim, Germany
Franziska Kluegl	University of Örebro, Sweden
Matthias Klusch	DFKI, Germany
Martin Kollingbaum	University of Aberdeen, UK
Ryszard Kowalczyk	Swinburne University of Technology, Australia
Jaroslaw Kozlak	University of Science and Technology in Krakow, Poland
Jiming Liu	Hong Kong Baptist University, China
Beatriz López	Universitat de Girona, Spain
Adolfo López Paredes	University of Valladolid, Spain
Zakaria Maamar	Zayed University, United Arab Emirates
Rene Mandiau	University of Valenciennes, France

Philippe Mathieu	University of Lille, France
Eric Matson	Purdue University, USA
Fabien Michel	University of Reims, France
José M. Molina	Universidad Carlos III de Madrid, Spain
Mirko Morandini	University of Trento, Italy
Jean-Pierre Muller	CIRAD, France
Jörg P. Müller	Clausthal University of Technology, Germany
Peter Novak	Czech Technical University in Prague, Czech Republic
Akhihiko Ohsuga	University of Electro-Communications, Japan
Eugenio Oliveira	University of Porto, Portugal
Andrea Omicini	University of Bologna, Italy
Sascha Ossowski	University of Rey Juan Carlos, Spain
Julian Padget	University of Bath, UK
Van Parunak	New Vectors, USA
Juan Pavon	Complutense University of Madrid, Spain
Michal Pechoucek	Czech Technical University, Czech Republic
Paolo Petta	University of Vienna, Austria
Michael Pirker	Siemens AG, Germany
Alessandro Ricci	University of Bologna, Italy
Juan Antonio Rodriguez Aguilar	AI Research Institute, Spain
Alex Rogers	University of Southampton, UK
Nicolas Sabouret	University of Paris 11, France
Silvia Schiaffino	ISISTAN, Argentina
Toshiharu Sugawara	Waseda University, Japan
Simon Thompson	British Telecom IIS Research Centre, UK
Paolo Torroni	University of Bologna, Italy
Rainer Unland	University of Duisburg, Germany
Domenico Ursino	University of Reggio Calabria, Italy
Jacques Verriet	Embedded Systems Institute, The Netherlands
José Villar	University of Oviedo, Spain
Jiri Vokrinek	Czech Technical University in Prague, Czech Republic
Gerhard Weiss	University of Maastricht, The Netherlands
Danny Weyns	Linnaeus University, Sweden
Niek Wijngaards	Thales, D-CIS lab, The Netherlands
Gaku Yamamoto	IBM, Japan

Organizing Committee

Juan M. Corchado (Chair)	University of Salamanca, Spain
Javier Bajo (Co-chair)	Polytechnic University of Madrid, Spain
Juan F. De Paz	University of Salamanca, Spain
Sara Rodríguez	University of Salamanca, Spain
Dante I. Tapia	University of Salamanca, Spain
Fernando de la Prieta Pintado	University of Salamanca, Spain
Davinia Carolina Zato Domínguez	University of Salamanca, Spain
Gabriel Villarrubia González	University of Salamanca, Spain
Alejandro Sánchez Yuste	University of Salamanca, Spain
Antonio Juan Sánchez Martín	University of Salamanca, Spain
Cristian I. Pinzón	University of Salamanca, Spain
Rosa Cano	University of Salamanca, Spain
Emilio S. Corchado	University of Salamanca, Spain
Eugenio Aguirre	University of Granada, Spain
Manuel P. Rubio	University of Salamanca, Spain
Belén Pérez Lancho	University of Salamanca, Spain
Angélica González Arrieta	University of Salamanca, Spain
Vivian F. López	University of Salamanca, Spain
Ana de Luís	University of Salamanca, Spain
Ana B. Gil	University of Salamanca, Spain
M^{a} Dolores Muñoz Vicente	University of Salamanca, Spain
Jesús García Herrero	Carlos III University of Madrid, Spain
Elena García	University of Salamanca, Spain
Roberto González	University of Salamanca, Spain

Table of Contents

Regular Papers

Demo Papers

Preventing Elderly from Falls: The Agent Perspective in EPRs

Sebastian Ahrndt*, Johannes Fähndrich, and Sahin Albayrak

DAI-Labor, Technische Universität Berlin, Germany
Ernst-Reuter-Platz 7
10587 Berlin, Germany
sebastian.ahrndt@dai-labor.de

Abstract. This work presents an approach combining multiple electronic patient records (EPR) to a self-learning fall risk assessment tool. We utilized the agent-perspective to model the system, to address privacy issues and to evaluate different distributed information fusion and opinion aggregation techniques towards there applicability to the addressed domain. Each agent represents a single patient negotiating about unknown fall risk influences in order to adapt the fall-risk assessment tool to the population under care. In addition, we will outline the planned real-world case study.

1 Introduction

Injuries or disabilities contracted by falls constitute not only as a cause of suffering for the elderly, but also lead to substantial economic burdens. For example, the total amount of fall related costs amount to 0.85% - 1.5% of the yearly German health care costs [14]. Here, the increasing development of electronic patient records (EPR) offers new opportunities for the prevention of falls as EPR contain a treasure trove of data [10]. Although, professionals gain benefits from the consolidation of such data sources, the available feature space is to huge to conceive for humans (e.g. the ICD-10 comprise more then 68,000 codes). In order to not overwhelm the user, EPRs provide standardized tools – such as fall-risk assessment tools – utilising a subset of the feature space in order to assist the user. In this work, we want to introduce such a fall-risk assessment tool which is agent-based and self-learning in order to reveal some kind of personal healthcare based on huge amount of data. To start with, a literature research done to identify well-researched fall-indicators and their influences on patients fall-risk were conducted [21]. Here, we were able to identify 25 fall-risk indicators. However, during the course of this work we applied the agent perspective to negotiate about unknown fall-risk influences in order to adjust the fall-risk assessment tool to the population under care. This requires population-based data, which can not be provided by studies as those conceived during the literature research as it would be impractical to conduct as much studies as populations exist [10].

* Corresponding author.

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 1–12, 2013.

Nevertheless, studies still outline the starting point for personalised health care. Consequently our approach utilised the identified fall-risk indicators and their influences as part of the initial knowledge of each agent – where each agents represents a single patient (See Section 2). Further, we applied the results of the literature review as part of the evaluation of our approach (See Section 3) and as default version of the assessment tool used during the planed case study (See Section 3.1).

The contribution of the work is threefold. To start with, we underline the applicability of contemporary agent-frameworks within EPR, due to their support for data privacy mechanisms. Germany in particular has strict rules when it comes to patient data, as such, privacy issues are not considered a nice extra, but required by law. Further, we show that constituting features of agents (e.g. the sensor-effector metaphor or cooperation) can be utilized to adjust EPR components to the population under care pro-actively. Subsequently, we compare different pooling methods for distributed information fusion and opinion aggregation in order to identify ways in which such adjustments can be established and, further, to identify the most-fitting method for the addressed domain.

2 Approach

In order to obtain the goal of personalized health-care there exist the need to observe population-based data and to utilise the observation results to adapt the original conceived health-care [10]. One can imagine, that this is a common procedure in the daily routine of heath-care professionals but a rather hard task for computers. During the course of this work, we want to outline an agent-based approach which achieves this for a fall-risk assessment tool as part of a mobile EPR [1]. Such tools enable health-care professionals to determine the individual fall-risk of the patients based on several fall-risk indicators and as a consequence to initiate suitable retaliatory actions. The basic idea of the intended approach is to enable the patients to negotiate about arising fall-risk indicators which are not yet available in the assessment tool. Here, each health-care professional takes care of several patients. We utilised the agent-based approach to model such a system as illustrated in Fig. 1. Consequently each patient is represented by a single agent, further, multiple patient agents run on a single platform representing a health-care professionals tablet. The hole environment consists of several tablets. However, Fig. 1 also illustrates that the negotiation process consists of four different stages to determine whether an adaption is necessary or not: The occurrence of a fall, the local information fusion, the global opinion aggregation and the notification stage. In the following we will explain each of the stages in detail. Afterwards we will present a comparison of different distributed information fusion techniques and outline the most fitting one for the addressed problem.

2.1 Elaborating Fall-Risk Indicators

Whenever a patient happens to fall, the incident is documented by the health-care professional. The health-care professional adds the observed fall-event $d \in D$

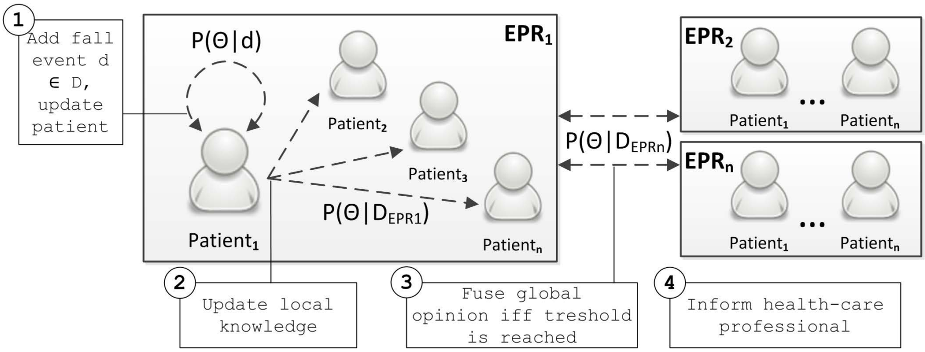


Fig. 1. Illustrating the approach where the patient agents at a single EPR negotiate which feature needs to be evaluated in negotiation with other EPRs and its appropriated patient agents. The process starts with the occurrence of a fall-event where the appropriated patient agents updates the probabilities of all features conceived (1). Afterwards the agent aggregates its new personal opinion with the prior local opinion of the node and propose the result to the other patient agents available through the node (2). If one feature reach a given threshold the agents aggregates the global opinion requesting all local opinion of the available nodes (3). If the threshold of the feature is still exceeded the EPR notifies the health-care professional about the new perceived fall-risk indicator (4).

to the EPR of the associated patient agent. With each observation d, new evidence is collected to elaborate possible fall-risk indicators in the feature space Θ, where Θ are all the quantities of interest on which the group wants to elaborate an opinion. In our domain the Θ represents the set of impact factors on a patients fall-risk, with $\Theta = \{\theta_i | i = 1..n\}$. For each patient the patients agent $p \in P$ elaborates at each observation the evidence for all fall-risk factors. The agent then updates its believe by using Bayesian information fusion [15] as illustrated by Eq. 1.

$$p(\Theta|d) = \frac{p(d|\Theta)p(\Theta)}{p(d)} \propto p(d|\Theta)p(\Theta) \qquad (1)$$

The a-priori $p(\Theta)$ represents the prior knowledge of the quantities of interest. As mentioned above we received this prior knowledge from a literature overview. Here we might exclude e.g. surnames of a patient record to be a fall-risk factor by setting $p(\theta_{surname}) = 0.0$. The likelihood $p(d|\Theta)$ represents the impact of the observation of d on the belief state of the agent. The a-posteriori $p(\Theta|d)$ can be calculated with these two inputs [2]. In this work the probability distribution $p(\Theta|d)$ can be seen as an experts opinion. As the observations $d \in D$ are assumed to be conditional independent, the expert opinion represents the 'degree of believe' of a single patient agent [11]. Each expert $p \in P$ has an degree of believe on which fall-risk factor has influenced the fall observed in the observation d and formulates this opinion in $p(\Theta|d)$. After each observation the expert updates its believe over Θ. One can easily imagine the vast amount of update functions possible. For example, Eq. 2 represents one update function which whether doubles the θ_i if the feature changed between this observation and the last one or halved it otherwise.

$$f(\theta_i) = \begin{cases} 2\theta_i & \text{if } d^t \neq d^{t+1} \\ \frac{1}{2}\theta_i & \text{if } d^t = d^{t+1} \end{cases} \quad (2)$$

With such an update function the likelihood of each feature can be calculated as shown in Eq. 3.

$$p(d|\theta) = \frac{f(\theta_i)}{\sum_1^n f(\theta_i)} \quad (3)$$

The fusion of the new evidence with each preceding observation is then integrated into the agents believe by using Bayesian information fusion as show in Eq. 4.

$$p(\Theta|d^{t+1}) \propto p(d^t|\Theta)p(\Theta|d^t) \quad (4)$$

After the agent has a new degree of believe $p(\Theta|d^{t+1})$ the first stage is completed. Now the local information fusion updates the group opinion of all patient agents on the same node. Here, in contrast to the update function of a single agent, if between agents the same θ is found, its impact factor should increase to exchange the effects of Eq. 2. Since the communication on one tablet is cheap and secure, the agents on each tablet are enabled to reassess the fall-risks after each observation of a fall locally. We need to introduce this distinction between intra and inter Tablet communication because the region the health-care professionals are working in is not fully covered with wireless Internet connections.

After the probability of a possible fall-risk indicator exceeds the experimentally established threshold shown in Eq. 5 the global opinion aggregation starts. Here the patient agents on one tablet fuse their node opinion with the more global opinion of all devices using the same method as in the local information fusion phase. If the threshold of the feature is still exceeded the suggested feature is forwarded to the health-care professional which has to decide whether this feature is a fall-risk indicator for the population under care or not. The result of this decision is communicated to all available nodes and if necessary updates the fall-risk assessment tool with the new feature. Each θ which was handled by a health-care professional is removed from the opinion elaboration by setting $p(\theta_k) = 0.0$. A side effect of the distributed information fusion is anonymisation. Since the probability distribution has been created through the aggregation of multiple opinions the communication between the different tablets during the global opinion aggregation becomes secure. The messages contain only the probabilities but not the values of the features as the values are not needed anymore at this stage.

$$\frac{1}{|\Theta \setminus \{\theta_k | p(\theta_k) = 0.0\}|} \quad (5)$$

Now each patient agent represents some kind of fall-risk expert which is able to express its opinion about the fall-risk influences of its patient. Following *Roback* and *Givens* the arising issue 'is to pool opinions in a rational way that yields a single probability distribution from which inference can be made' [19]. Here, we might use methods of information fusion to establish a group opinion [22].

2.2 Aggregating the Group Opinion

As usual in the information fusion we need some kind of aggregation method to fuse these local expert opinions together to a generalized view [23]. Since there are multiple options, we have to choose an appropriate method (a so called pooling method) to combine multiple opinions into a group opinion. Pooling methods might represent different voting strategies like an dictatorship of one opinion, a democracy where every agents opinion has the same weight or an strategy which is based on the reliability of the experts. The interested reader is referred to *Faehndrich* [4].

However, to see how those pooling methods work, we will look at some of them, and evaluate their usefulness in our multi-agent system. Some of the best known pooling methods are the 'Linear Opinion Pool' [7] (LinOp), the 'Logarithmic Opinion Pool' [9] (LogOp) and the 'Supra-Bayesian Pooling Method' [8]. Each of them profits from a growing body of evidence [3]. There are two ways on evaluating a pooling method: On the one hand, we can evaluate their theoretical properties like a 'non dictatorship' or an 'unanimity' [18]. On the other hand one can measure their performance in a real world example. In this work, we want to evaluate how those theoretically well-researched pooling methods can be used in the real world task of elaborating fall-risks as a group decision in a multi-agent system.

Table 1. Classification of the examined pooling methods and there applicability to the domain we address. With n being the amount of patient agents and m being the number of nodes (devices) available.

Method	Space	Time	Communication
Supra-Bayesian	$O(1)$	$O(1)$	$O(m+n)$
Linear Opinion Pool	$O(m*n^2)$	$O(m*n^2)$	$O(m+n)$
Logarithmic Opinion Pool	$O(m*n^2)$	$O(m*n^2)$	$O(m+n)$

Table 1 classifies the examined pooling methods utilizing several criteria. Here we point out the time, space and communication complexity of the pooling methods. Although, the opinion aggregation using LinOP and LogOP requires the mean value and therefore requires to aggregate the opinion of every agent and node depending on the voting round, the communication complexity of all pooling method is equal. This is due the fact, that using the Supra-Bayesian the agents have to broadcast the new aggregated opinion to all other agents and nodes as illustrated in Fig. 1. However, the time and space complexity differs. Since we have two voting rounds, one local on the node and another one between nodes, the space and time complexity rises quadratically using LinOP or LogOP. Here, the Supra-Bayesian Pooling Method can reuse the last available pooling result generated in the prior voting round to update the node and/or group opinion. This reduces the time and space complexity to be linear.

3 Evaluation

In order to evaluate the described approach we developed a prototype using the agent framework JIAC V [12]. Here, we decided to conduct a simulation in order to test the algorithm. Each patient agent where set-up with the initial set of well-researched fall-risk indicators which we received from our literature research [21]. All utilized features such as age, sex or diseases are available in contemporary EPR and can be ordered through their impact on the fall-risk of a patient. For example, a physical deficit in the lower extremities influences a patients fall-risk more than the age of the patient (factor 4.4 vs. 1.7) [20]. We implemented the patient agents as described in Section 2 and migrated the data model of the EPR developed within the agnes^{zwei} project [1]. Further, we implemented an agent (SimAgent) which simulates the health-care professionals role. The SimAgent adds a fall-event each 100ms to the multi-agent system and decides whether a suggested feature is a fall-risk indicator for the population under care or not. As fall-events do not occur randomly the SimAgent decides which patient should fall based on the patients current fall-risk. Further, the SimAgent changes the features of the patient record as it would be during the use of the EPR system. The simulation of the aging process of a patient is based on the research results of prior studies with risk-equivalent patients [5,6,13]. We utilized this behavior to evaluate the system. In the following we will discuss the simulations result of each of the three implemented pooling methods. For each pooling method we observed 50 features during the first 80 voting rounds and carried out several simulation runs.

Despite that the simulation always starts with the same initial set of patients, the aging process is non-deterministic and varies between the simulation runs. Therefore the subsequently presented figures can not be compared directly.

Fig. 2 illustrates the probability distribution for the LinOP method. To aggregate the opinions of the patient agents LinOP requires weights for each expert. During the simulation we set the weights to $\frac{1}{|Agents|}$ which can be interpreted as a fair democracy. Here we observed that the probability of the observed features grows slowly during the first 80 voting rounds. LinOP did not exceed a maximum of 0.14 for a single feature making it necessary to observe a greater number of falls. In our real-world problem we do not expect this high rate of falls.

Fig. 3 illustrates the probability distribution for the LogOP method. For the simulation with LogOP we applied the same weights as used in the LinOP simulation. During the first voting rounds LogOP produces similar results as the

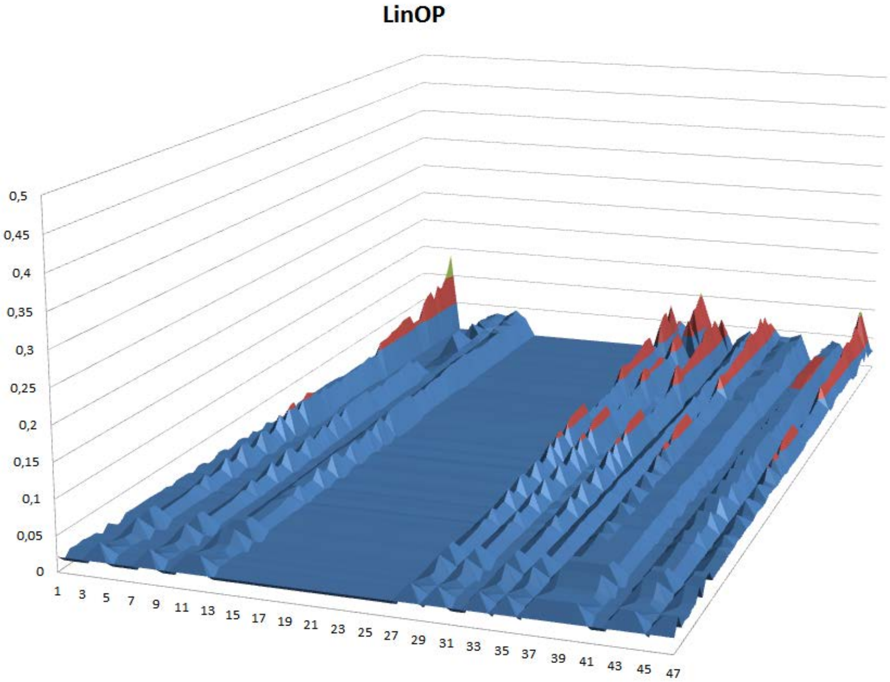


Fig. 2. Progress of the probability distribution using the LinOP method. The x-axis shows the feature, the y-axis the probability and the z-axis the voting round.

LinOP does. The group decision behavior changes at the moment the first features reach a zero probability. Since with a growing body of evidence single features get excluded. One must notice, that in the case of LogOP a single expert vote with $p(\theta) = 0.0$ suffices to exclude a single feature disregarding all other expert opinions. This is based on the multiplicative nature of the pooling method making single feature disproportional likely.

Fig. 4 illustrates the probability distribution for the Supra-Bayes Pooling method. Using Supra-Bayes it is difficult to decide which likelihood function to use after an observation d. For the simulation we applied the arbitrary function shown in Eq. 2. Supra-Bayes requires no weights presenting a rational way of pooling a democratic group opinion. In contrast to LogOP no single expert can use its vote to disregard all other votes. In addition, Supra-Bayes profits much faster from a growing body of evidence. The first feature exceeds the threshold after only a few voting rounds. This can be seen as advantage and disadvantage. For our real-world problem we profit from the fast convergence since less falls have to be observed.

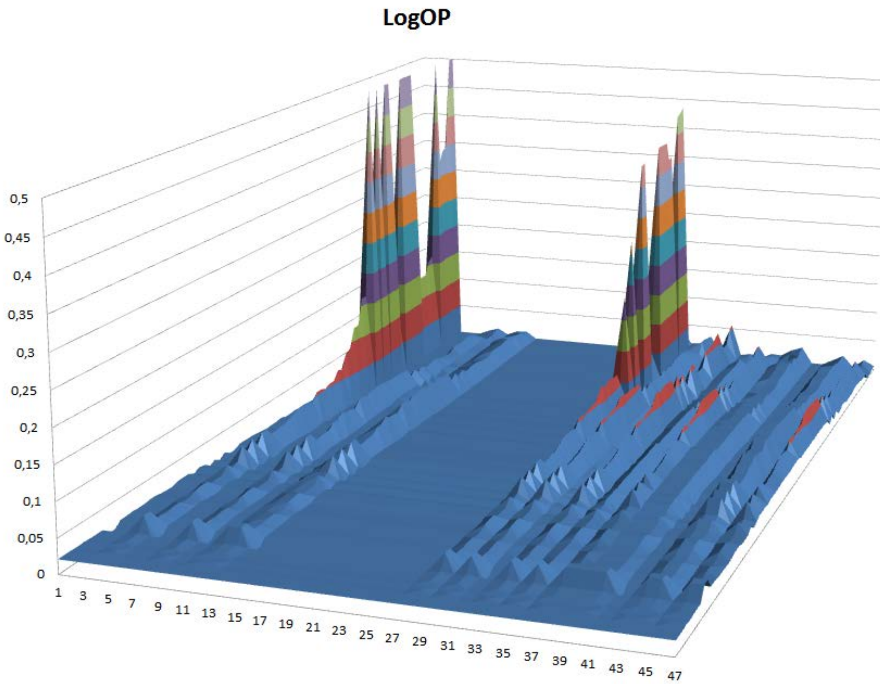


Fig. 3. Progress of the probability distribution using the LogOP method. The x-axis shows the feature, the y-axis the probability and the z-axis the voting round.

3.1 Field Test

The simulation results presented above showed that our approach seems valid and is applicable to the problem it addresses. In addition, we are currently facing the launching date of the real-world field test. The first phase of the field test includes six home-visiting nurses engaged in the wide-spreaded area of Brandenburg, Germany. This nurses will treat between 180 to 300 patients which are at least 65 years old and multi-morbid (more then three diseases). The first phase is scheduled for three month and we expect a noticeable number of fall events as prior studies with risk-equivalent patient groups show that the addressed patients have a rather high fall-risk (up to 60% per year [5]). As this rate only addresses the self-reported fall events of the patients it will be interesting to see how many falls actually occur.

3.2 Selecting a Pooling Method

To conclude, for our approach we finally decided to use the Supra-Bayesian method through the following reasons.

During our simulation run we analysis the first 80 voting rounds where each voting round was triggered by a single fall observation. During the field test

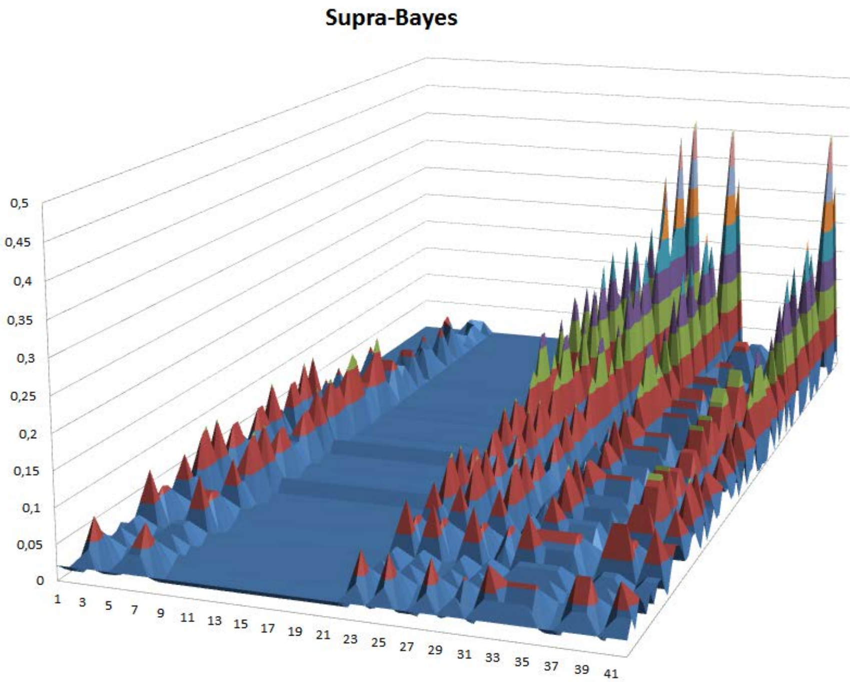


Fig. 4. Progress of the probability distribution using the Supra-Bayesian method. The x-axis shows the feature, the y-axis the probability and the z-axis the voting round.

we expect less falls. As mentioned above the health-care professional will treat between 180 to 300 patients. Therefore, we can calculate that we will observe approximately 30 to 50 falls. Only the Supra-Bayesian Pooling method is able to produce meaningful results under this circumstance. Hence, one reason is the expected fall-rate. Another reason is the space and time complexity as the addressed platform are tablets and the system should act resource-efficient. Even the communication complexity remains equal for all considered pooling methods, the Supra Bayesian method outperforms the others in time and space complexity here (as shown in Tab. 1). In addition, we can underline the finding of *Roback* and *Givens* [19] that the implementation of LinOP and LogOP in real problems was not simple as weights for the group members or the likelihood function have to be chosen. As this might be an interesting opportunity for future work (weighting of experience), this effort is not necessary for the current field test.

4 Conclusion

In this paper, we introduced an agent-based approach for a self-learning fall-risk assessment tool. This tool as part of a mobile EPR enables health-care professionals to determine the individual fall-risk of the patients based on several fall-risk indicators. Usually this fall-risk indicators are based on studies which are not able to capture population based data. In order to adapt the assessment tool to the population under care, we enabled the patients to be part of the elaboration of arising fall-risk indicators. Here each patient is represented by a single agent where each mobile EPR contains multiple agents. The whole environment consists of multiple EPRs. In order to aggregate all opinions we introduced an approach consisting of four steps: The occurrence of a fall, the local information fusion, the global opinion aggregation and the notification stage. During the local information fusion and the global opinion aggregation each patient votes as a fall-risk expert. Here, we applied three different information fusion methods to pool a rational single opinion, where a single opinion represents a probability distribution over all features in the feature space. The evaluation of the fusion techniques emphasises that one method outperforms the others under the special circumstance of our application. Here, we showed that the fall frequency we expect is not large enough for the other methods to produce meaningful results. Also we require a method which is resource-efficient in time and space complexity as the addressed platform are tablets.

4.1 Future Work

The presented approach utilises a great amount of information available through the EPR. However, at this point of development the information fusion is limited to data sources which can be easily computed by the agents. Even the software engineers and designers try to standardize and automate most of the binary input in EPRs, there will be always the need to provide free text fields for the health-care professionals as the addressed working environment is to manifold to normalize it. Hence, in the next stage we want to make accessible the full potential of EPRs using Natural Language Processing techniques to evaluate the free text fields. Furthermore, we plan to integrate external influences into the risk decision process. Which means that we want to examine if weather conditions or the actual season of the year also influence the fall risk of elderly. Another interesting research focus in information fusion for EPRs is the integration of sensor data into health records (e.g. *Mohomed et al.* [16] and *Moulton et al.* [17]). Although the aggregation of sensor data is not in the focus of this work it provides interesting aspects for the future with upcoming Bluetooth 4.0 health devices. Extending the research on pooling methods, different pooling methods have to be evaluated as well as their parameters. Analyzing different update function and impact on the learning algorithms will be further focused.

References

1. Ahrndt, S., Rieger, A., Albayrak, S.: Entwicklung einer mobilen elektronischen Patientenakte für die ambulante Versorgung in ländlichen Regionen (development of a mobile electronic patient record for ambulatory health care in the countryside). In: Goltz, U., Magnor, M., Appelrath, H.J., Matthies, H., Balke, W.T., Wolf, L. (eds.) INFORMATIK 2012. Lecture Notes in Informatics, vol. (208), pp. 1167–1181. Gesellschaft für Informatik, Braunschweig (2012)

2. Bernardo, J.M., Smith, A.F.M.: Bayesian theory. Wiley series in probability and statistics, Wiley, Chichester [u.a.], repr. edn.(2004), http://www.ulb.tu-darmstadt.de/tocs/133503178.pdf
3. Cooke, R.: Experts in Uncertainty: Opinion and Subjective Probability in Science. Oxford University Press (January 1991)
4. Fähndrich, J.: Analyse von Verfahren zur Kombination von Expertenwissen in Form von Wahrscheinlichkeitsverteilungen im Hinblick auf die verteilte lokale Bayes'sche Fusion. Diploma thesis, Karlsruhe Institut of Technology (May 2010)
5. Fiss, T., Dreier, A., Meinke, C., van den Berg, N., Ritter, C.A., Hoffmann, W.: Frequency of inappropriate drugs in primary care: Analysis of a sample of immobile patients who received periodic home visits. Age and Ageing 40(1), 66–73 (2010)
6. Fiss, T., Ritter, C.A., Alte, D., van den Berg, N., Hoffmann, W.: Detection of drug related problems in an interdisciplinary health care model for rural areas in germany. Pham. World Sci. 32(5), 566–574 (2010)
7. Genest, C.: Pooling operators with the marginalization property. The Canadian Journal of Statistics/La Revue Canadienne de Statistique 12(2), 153–163 (1984), http://www.jstor.org/stable/3315179
8. Genest, C., McConway, K.K., Schervish, M.M.: Characterization of externally bayesian pooling operators. The Annals of Statistics 14(2), 487–501 (1986), http://www.jstor.org/stable/2241231
9. Genest, C., Weerahandi, S., Zidek, J.: Aggregating opinions through logarithmic pooling. Theory and Decision 17, 61–70 (1984), http://www.springerlink.com/index/XT152131QJ43G001.pdf
10. Goth, G.: Analyzing medical data. Communications of the ACM 55(6), 13–15 (2012)
11. Halpern, J.: From statistical knowledge bases to degrees of belief: an overview. In: Proceedings of the Twenty-Fifth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, p. 113. ACM (2006), http://portal.acm.org/citation.cfm?id=1142367
12. Hirsch, B., Konnerth, T., Heßler, A.: Merging agents and services – the JIAC agent platform. In: Bordini, R.H., Dastani, M., Dix, J., Amal, E.F.S. (eds.) Multi-Agent Programming: Languages, Tools and Applications, pp. 159–185. Springer (2009)
13. Hoffmann, W., van den Berg, N., Thyrian, J.R., Fiss, T.: Frequency and determinants of potential drug-drug interactions in an elderly population receiving regular home visits by GPs – results of the home medication review in the AGnES-studies. Pharmacoepidemiology and Drug Safety 20(12), 1311–1318 (2011)
14. König, H.H.: Gesundheitsökonomische Aspekte der Sturz- und Frakturprävention (2012), http://www.sturzpraevention2012.de/programm_abstract.php?No=9 (last visit: October 29, 2012)
15. Lindley, D.: Theory and practice of bayesian statistics. The Statistician 32(1), 1–11 (1983), http://www.jstor.org/stable/2987587
16. Mohomed, I., Misra, A., Ebling, M., Jerome, W.: HARMONI: Context-aware filtering of sensor data for continuous remote health monitoring. In: Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2008), pp. 248–251. IEEE Computer Society (March 2008)
17. Moulton, B., Chaczko, Z., Karatovic, M.: Data fusion and aggregation methods for pre-processing ambulatory monitoring and remote sensor data for upload to personal electronic health records. International Journal of Digital Content Technology and its Applications 3(4), 120–127 (2009)

18. Pennock, D., Wellman, M.: Graphical models for groups: Belief aggregation and risk sharing. Decision Analysis 2(3), 148–164 (2005), http://ai.eecs.umich.edu/people/wellman/pubs/decisionanalysis05.pdf
19. Roback, P.J., Givens, G.H.: Supra-bayesian pooling of priors linked by a deterministic simulation model. Communications in Statistics – Simulation and Computation 30, 447–476 (2007)
20. Rubenstein, L.Z., Josephson, K.: Falls and their prevention in elderly people: What does the evidence show? Med. Clin. North Am. 90(5), 807–824 (2006)
21. Schenk, A., Ahrndt, S., Albayrak, S.: Predicting fall risks in electronic patient records. In: Goltz, U., Magnor, M., Appelrath, H.J., Matthies, H., Balke, W.T., Wolf, L. (eds.) INFORMATIK 2012. Lecture Notes in Informatics, vol. (208), pp. 1194–1198. Gesellschaft für Informatik, Braunschweig (2012)
22. Torra, V., Narukawa, Y.: Modeling decisions: information fusion and aggregation operators. Springer (2007)
23. Wark, E.B.J.S.: Concepts, Models, and Tools for Information Fusion. Artech House (2007), http://www.amazon.com/Concepts-Models-Tools-Information-Fusion/dp/1596930810

Dynamic Organisation of the Household Activities for Energy Consumption Simulation

Edouard Amouroux[1] and François Sempé[2]

[1] LIP6 - Université Pierre & Marie Curie, Paris, France
[2] François Sempé AE, Paris, France

Abstract. Taking into account inhabitants' activities is a necessity for efficient power management, especially in a residential context. In this paper we present an agent-based modelling and simulation framework allowing precise description of the inhabitants behaviour. This framework provides household representation with dynamic organisation capabilities. After introducing the proposed processes, we demonstrate the capabilities of the system to represent coherently the household dynamic organisation in terms of constrained and emergent habits and response to unforeseen event (new electricity tariff and change of the household activities).

Keywords: Agent-Based Modelling, Social Simulation, Electricity Consumption, Inhabitants' Behaviour Adaptation.

1 Introduction

Energy management is becoming an important trend in our society due to greenhouse effect concerns and growing tensions on energy markets. A large amount of the CO_2 emission is due to residential building electricity consumption as it represents around 27% of the final energy used in Europe [2] and 37% in the USA [14]. Power management in the residential sector faces two main challenges: (1) power distribution and balancing and (2) peak consumption reduction. The first is mostly addressed through the development of the smart grids [6] which enables real-time response of electricity production according to the actual demand. In order to meet peak consumption demand, ESCOs (energy supplying company) and DNOs (distribution network operators) are forced to oversize their production and distribution capabilities. Such peaks are short and rare but have a large impact on the overall system. Certainly, the most effective way to limit the peak consumption issue is to change household consumption patterns [13]. Some studies are being conducted to evaluate the efficiency through awareness programs or by the set-up of smart meters allowing inhabitants to track in real-time their electricity cost and the induced carbon emissions [8]. Nevertheless, these field studies are time demanding and costly to conduct, simulation with explicit representation of inhabitants behaviours and how they respond to various incentives would be a solution to such issues.

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 13–24, 2013.

We develop the SMACH modelling and simulation framework (see [1] for a complete description) to enable field-experts to investigate inhabitants activities and their relation to the household electricity consumption. In the present paper, we present the organisation processes added to SMACH that allows adaptation of the household activity in response to unforeseen events such as change of electricity tariff or modification of the household constraints.

The rest of the paper is organised as follows. Section 2 briefly presents how household consumption and its change have been modelled so far and then introduces the key elements of the SMACH meta-model. Section 3 describes precisely the two organisation processes proposed in SMACH. Section 4 demonstrates the dynamic organisation of the household with several simulation examples. Section 5 concludes the paper and introduce the current perspectives which are related to improving environment's diversity and complexity and better calibration and validation using upcoming field data.

2 Behavioural Model for Energy Consumption Simulation

Residential electricity consumption has been largely studied from a top-down approach in which the sector is considered as a whole [12]. In contrary, few bottom-up proposals, which consider the household unit, have been made.

2.1 Existing Bottom-Up Approaches

A first class of bottom-up models is the statistical one which use energy consumption data and match them to macroeconomic indicators in order to generate average consumption prediction. The second one, called "engineering method" (EM), uses the dwelling characteristics and appliances' power ratings to generate more individualised prediction. Even though, EM models are clearly more adapted to investigate household energy consumption patterns, we agree with [5] that such models are not sufficient due to their static nature (*i.e.* static consumption profiles).

Whereas more than a few agent-based models have been proposed to study the electricity markets (considering the interactions between ESCOs, DNOs and consumers) [15], limited attempts to represent household inhabitants have been made. In [5], the authors propose to represent explicitly the inhabitants behaviours through a reactive and deliberative decision processes.The authors evaluated numerous deliberative agent architecture and selected the Brahms one [11] as it is similar to the BDI architecture (belief-desire-intention) and its clear definition of thought-frames and work-frames. Nevertheless, we argue that such complex representation of the decision process is not necessary [3] to obtain a model able to tackle our objective: investigating adaptation of household activities to various incentives and events. Moreover, using cognitive-oriented decision are likely to impede the design of experiments by field. In contrary, our activity-oriented model do not require knowledge about human cognition but remains adapted to represent both individual an collective actions.

2.2 SMACH

The key elements of the SMACH meta-model related to the adaptation mechanisms are presented in this section. A more detailed description of the meta-model and the associated simulator can be found in [1].

Metamodel. The environment is limited to the household which is a set of rooms where appliances and inhabitants are located (plus a special location for all outdoor activities). An appliance is characterised by a set of states or a set of programs which defines its electrical consumption. The appliances' electrical consumption are drawn from field measures conducted during the REMODECE european project [1]. In addition, a current date and an electricity tariff are globally defined.

The inhabitants activity is decomposed into generic *tasks* (t) such as *watch TV*, *cook dinner*, etc. The behaviour of each individual is a set of *actions* (a) derived from the generic tasks. A task is a tuple $\langle \tau_{min}, \tau_{max} E_t, E_{tf}, T_{pre} \rangle$ corresponding respectively to minimal and maximal duration to conduct the task, the required and favourable appliances and the pre-conditional tasks. For instance, *ironing* may require the completion of *cleaning clothes*, lasts one to two hours, requires the iron appliance and may be likely to be conducted with the TV switched on.

An action is defined by a tuple $a = \langle t, w, st \rangle$ corresponding respectively to the associated task, rhythm and its conduction state $(done, not_done)$. A *rhythm* (w) is a tuple $\langle per_w, freq_w, var_w, Pp_w \rangle$ corresponding respectively to the base period (day, week, month, year), the frequency, the frequency variability and a set of preferred period (PP). An action example could be as follows: a child may conduct the *watch TV* action according to a weekly rhythm of 10 realisations to be conducted between 7 to $8 a.m$ and/or 5 to 6 $p.m.$

Whereas the rhythm represents inhabitants' habits, the *event* mechanism represents exceptional situation. Concretely, an event replaces the rhythm of several action for a given period. More formally an event q is a couple $([d_{s_q}, d_{e_q}], W_q)$: a period and a set of couples (a, w) (action and new rhythm). For instance, a *sickness* event may prevents an individual to go to work (null frequency rhythm) and force the *resting* action for a few days.

In addition to the individual actions presented, some action can be conducted collectively and/or can benefit the whole household (*prepare dinner* for instance). This is represented by adding a *coll* and a $benef$ parameter to the task characterisation. This collective aspect of the household activity is made possible by a communication protocol based on the speech-act theory [9] allowing agents synchronisation (more details such as the performatives list can be found in [1]).

Action Selection. At any given time-step each inhabitant agent conducts exactly one action which is selected as follows. The agent filters the *possible* actions and then select the one with the highest priority level (resulting in continuing

[1] `http://remodece.isr.uc.pt`

the current actions or starting a new one). An action is considered *possible* if every pre-conditions task have been conducted and all the required appliances are available. The action priority is computed after two types of influence.

- *Agent internal state* influences: commitment to the current action, associated rhythm (*e.g.* is the current date within a preferred period?)
- *Environment* influences: current electricity tariff, other agents invitations (communication) and events

Let us note that only the relative order of the influences values is significant. For instance, an invitation influence higher than the commitment one means that invitations will interrupt current actions *ceterit paribus*. In addition, their initial values have been determined empirically though some of them are subject to an adaptation mechanism, throughout the simulation, that tends to force the conduction of action according to their rhythm. Finally, the priority of an action a can be boosted by another action a' having a as a pre-condition (*e.g.* the *work* action can boost the *take breakfast*'s priority).

3 Dynamic Organisation Process

In this section, we introduce the adaptation mechanisms that complement the previously presented meta-model and enable a dynamical organisation of the household activity. We use the term of organisation to describe the capacity of individuals to coordinate their behaviour in order to achieve a common goal. In our case, the goals of the inhabitants are represented by the rhythms of the actions - frequency and schedule constraints - in other words what people need or want to do every day and every week. This ability needs to be implemented as a continuous process in order to take into account unpredictable events like a new energy pricing or a modification of the household activity. Moreover, because it is continuous, the process has to be light too in order to be able to conduct long term yet precise simulation (*i.e.* our simulation lasts up to several and have a minute time-step). The dynamic organisation process is tow-fold: the first part deals with constrained habits and the second one manages the unscheduled actions.

3.1 Constrained Habits

The organisation process can be seen as a competition between actions for the satisfaction of both constraints. For the frequency constraints (*i.e.* doing what needs to be done), the competition is relative to the duration of a day in which a limited amount of actions can take place. For the schedule constraints (*i.e.* doing it when it should be done), the competition exists when several actions have overlapping PPs (preferred period). The time of realisation of an action is controlled by a specific influence on the action priority. This influence is positive or null during a PP (the bonus) and negative or null outside a PP (the penalty). In other words, the bonus and the penalty govern the likeliness for an action to be triggered inside or outside its PPs. At the end of a day, every action estimates its own satisfaction, and adapts its bonus and penalty according to the following policy :

Bonus increase : increase the probability of realisation during a PP. Used when the action lacks of realisations.

Bonus decrease : decrease the probability of realisation during a PP. Used when the action constraints have been satisfied for a while. It gives the opportunity for other actions to win the competition and thus ensure a slow re-organisation capacity.

Penalty increase : decrease the probability of a realisation outside a PP. Used when realisations happen outside a PP.

Penalty decrease : increase the probability of a realisation outside a PP. Used when bonus increase did not permit to reach the goal frequency, as we give priority to frequency constraint over PP constraints.

Bonus and penalty values are bounded (minimum is zero and maximum is set to respect the influence order see Section 2.2. In case of persistent failure for the realisation of an action, precisely when the bonus is already at its maximum and the penalty at its minimum, the PPs may be slightly extended.

3.2 Emergent Habits

Actions with no initial schedule represent activities that can be conducted at anytime... during the spare time left by scheduled actions. Those mobile actions need to be organised, that is to be given a schedule, for two reasons: scheduling improves the efficiency of the organisation and it is more realistic to give habits to the agents. Nevertheless, those emergent habits should be flexible to adapt themselves to the circumstances. In addition, the modeller is offered the possibility to force an action to remain unscheduled.

The second organisation process aims at creating flexible PPs for mobile actions. Its mechanism relies on the memorisation of the last realisation periods. Those periods are used to update the current schedule of an action. In other words, if one action find the opportunity to be triggered at a given time of the day - wether or not inside a PP, this period will be then favoured as long as it remains in the memory. It is particularly useful for coordinated and exclusive tasks thanks to the emergence of common habits for the first ones (instead of relying on invitations) and differentiated habits for the second ones (resulting in less conflicts.

In order to facilitate the mobile actions to find a first schedule, they receive no PP influence during the first week . Then, from time to time, on a random basis, a given action is *liberated* for a few days (*i.e.* its PP influence is set to zero). Those temporary suppressions of habits gives the action the opportunity to find a more suitable schedule or, another action to take the freed schedules. The frequency and the duration of those *liberations* are some of the factors that control the global mobility of actions, making the routine of a household more or less tight.

Thanks to these adaptation mechanisms, a household can dynamically organises its activities in relation to the current circumstances. The next section demonstrates such ability facing energy pricing changes and major modifications of the household routine.

4 Dynamic Organisation of a Household

In order to evaluate the household organisation capabilities, we introduce the *completion rate* that is the number of actual realisations over the number of realisations specified by the modeller. The difficulty - or even the possibility - to achieve a 100% completion rate depends on several factors including the number of coordinated and exclusive activities, the complexity of the schedule constraints and the *occupation level*: the time needed to achieve all the requested activities over the time available. A second indicator, *PP fulfilment rate* , of the organisation measures the fulfilment of the schedule constraints. In the present case, there is no conflicting schedules situations but such situation is presented and solved in [1].

In this section, 3 situations will be tackled: (1) Initial organisation process (2) Re-organisation in relation with a new energy pricing and (3) Re-organisation following a major change in the household activities. Firstly, let us describe the simulation setting.

4.1 Simulation Setting

The simulation involves a family of 2 parents (John and Mary) and a teenager (Bill). Each one of them receive 3 kind of tasks:

Scheduled tasks : *Sleeping* at night, taking the 3 main meals at the usual time, *Taking a shower* in the morning and *Going to work* or *to school* during the day from monday to friday, with the exception of wednesday for Mary and Bill. In the last simulation, Mary will have a different working schedule.

Mobile tasks : Those tasks have a frequency constraint but can be done at anytime during a a week. In this category, we found housework tasks and leisure tasks. The first ones are shared and can be done indifferently by one of the parents. They include *Cleaning* (using the vacuum cleaner), *Washing clothes* (feeding the washing machine and starting it) and *Ironing* (using the iron). The leisure tasks regroup different types of activities: some of them are coordinated (*Playing chess...*), just more collective than individual (*Watching TV*), free or even exclusive (using the computer for its own purpose.) They can be indoor or outdoor like *Hiking*, and require or not an electrical appliance.

Unconstrained tasks : those tasks have no constraints at all and thus have a low priority. They are triggered when no other tasks is available.

When considering the frequency constraints and the average duration of the actions, we obtain an occupation rate of $\approx 88\%$ per week for each inhabitant. It is sufficiently high to test the organisation abilities of our model.

Let us note that frequency and schedule constraints can be mitigated. For instance, a weak frequency constraint task will receive a lower influence than another one with a strong frequency constraint, thereby creating a hierarchy in the probability of realisation between the tasks. Although, in the following examples, all constraints have been made strong for the sake of clarity.

4.2 Organisation Building and Stability

The first simulation lasts 15 weeks. The figure 1 shows its completion rate by week. It is a typical outline as the completion rate increases during the first 3 or 4 weeks and then is relatively stabilised. The increase stage demonstrates the organisation process: unscheduled tasks find a proper settlement and scheduled tasks adapt their influences too, if necessary.

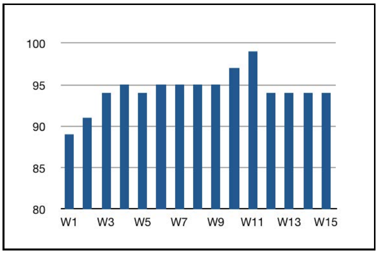


Fig. 1. Completion rate in % by week

In our case, the completion rate stabilises itself around 94% in average. The 6% missing realisation includes uncompleted realisations - started but interrupted before finished - (around 3%) and conflicting tasks. For instance, during 5 weeks, the family goes hiking on sunday around lunch time, thus missing the lunch. Let us precise that the permanent re-organisation process makes the missing realisations to move from one activity to another.

What should we think of the 94% completion rate? On the one hand, considering the occupation rate of 88% and the 6% of time spent in unconstrained activities - a time that could have been dedicated to constrained actions - we can assert that a better organisation, producing a higher completion rate, is possible. On the other hand, optimisation is not our concern: even though people organise their schedule, they do not completely optimise it. The first-weeks increase of the completion rate demonstrates an organisation building process, that is what we need to observe the household adaptation to new situations.

For a more concrete illustration of the organisation process, we will use the long-term activity diagram. This diagram is composed of a succession of coloured columns, one column for one day, and one colour for each task. On the horizontal axe, days are following one another. The long-term diagram offers a synthetic view of habits and change of habits over long periods.

The figure 2(a) shows the long-term diagram for 2 coordinated leisure tasks : *Gaming* in red (4 times a week for Bill and Mary with the use of a computer) and *Hiking* in yellow (2 times a week collectively) and a constrained one: the large purple strips represent *Going to school*. The latter helps us to identify wednesdays and week-ends (no purple stripe).

One can observe that *Going to school* with its strong constraints exhibits a very regular pattern. On the contrary, the 2 leisure tasks settle quickly on some specfici days and times, but move occasionally. This mobility may have several causes:

1. Due to the organisation process, an action may take the place of another one. For instance, at the beginning of the simulation, one of the gaming session takes place on tuesday night and prevents to have dinner (see figure 2(a)). For this reason, *Having diner* adapts its bonus and then, from the third week, it is able to interrupt the gaming session (the 2 separated red stripes).
2. Several stochastic factors intervene during a simulation. One of them is the draw of the number of realisation for an action on a given day. This is why the family who starts to go hiking on saturday afternoon switch from the third week to the morning. The previous weeks, they used to watch TV together. That day, Bill did not draw a realisation for *Watching TV*. Then, the priority of *Hiking* is high enough to be triggered for him and an invitation to hike to his parents is made. The invitation influence positively the parent in favour of this task and thus, they start to do it.
3. Some events like a new pricing or a change in the usual activities may disturb the organisation. We will address those situations in the next sections.

It is worth noticing that the organisation process produces a great deal of variability as shown in figure 2. Indeed, even if two families are very similar in terms of structure and constraints, they are unlikely to come with identical organisation (*i.e.* there are different solutions). Nevertheless, very constrained (*i.e.* scheduled) task, such as work or school, are conducted in a very similar organisation.

4.3 Reorganisation Following a New Energy Pricing

Here, we would like to illustrate a household reorganisation following a new energy pricing. In a simulation with the same settings, the energy price is uniform except in February. During this month, the power company decide to set a new pricing, with a higher price between 6-am and 12-am from monday to saturday. They may want their client to lower their consumption during this period.

As described before, a money saving trend has been added to our model as an action priority influence. More precisely, the energy consuming tasks receive a penalty during the costly periods and a bonus outside. The level of those added influences may vary from one agent to another in order to represent various attitudes in relation with money saving. In this simulation, Bill is less concerned than his parents.

The figure 3 shows the activity diagram for all the energy consuming tasks, with the grey squares indicating the costly periods. One can notice that most

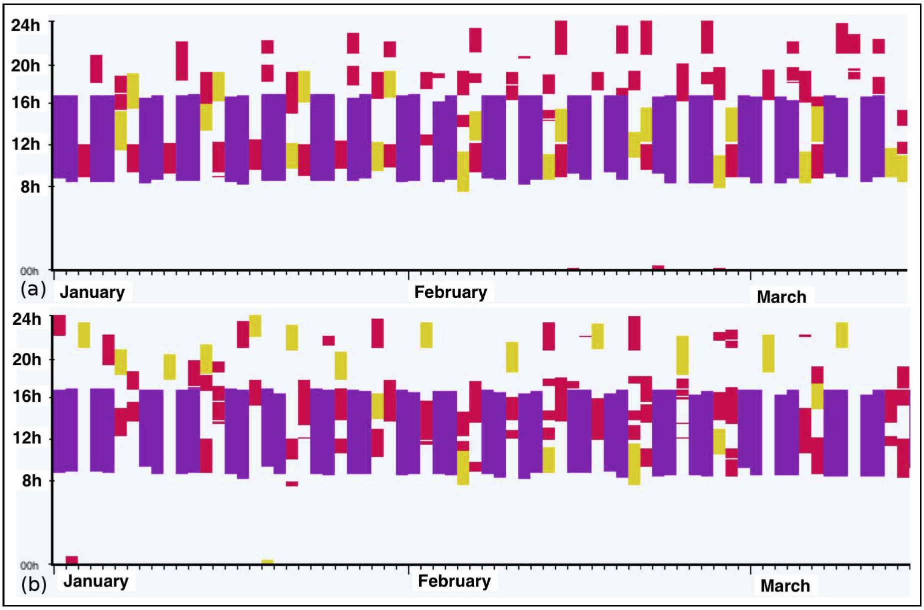


Fig. 2. Activity diagrams of 3 tasks: Going to school (purple), Gaming (red) and Hiking (yellow). (a) and (b) are two different simulation runs.

of the morning consuming tasks migrates to the afternoon and the evening from the beginning of February. Only the sunday morning, when the price is normal, keeps its routine. There is an exception: Bill still uses the computer on wednesday morning; the avoidance of costly periods can be personalised and tuned.

The migration of energy consuming tasks leads, necessarily, to a significant re-organisation of the activities. For instance, the gaming session of wednesday morning is displaced to the wednesday evening. Nevertheless, the disturbance is barely noticeable in term of completion rate: it is decreased by only 1% during the first week of february (week 5).

After the end of february, the price is uniform again. It takes about 3 weeks for the consuming tasks to populate again the morning. This is the time needed to overcome the habits taken during the month of February.

4.4 Reorganisation Following a Change in Household Activity

People have babies, go on holidays, get unemployed, etc.: major changes occurs in a household from time to time and a long term behavioural model have to deal with them. Our last example is a similar simulation, lasting 10 weeks, and with a change in Mary's activities: she starts the simulation unemployed and then, from the 5th week, go to work full time from monday to friday.

The completion rate of this simulation (figure 4(a)) shows a significant drop when Mary starts to work (from 97 to 93%). This drop comes from the time-occupation rate increase caused by the new workload of Mary. Indeed, there

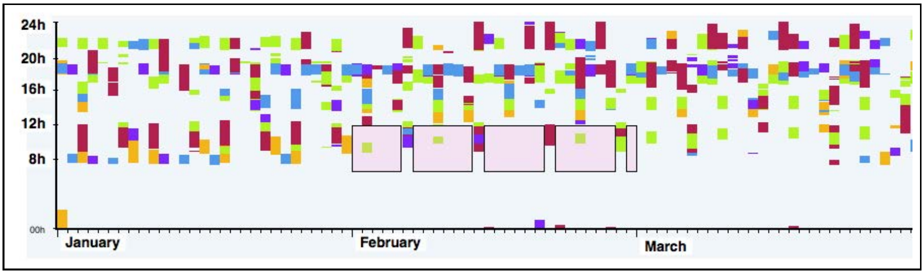


Fig. 3. Activity diagram with the tasks using electrical appliances. Grey squares show periods with higher energy prices.

is a direct link between the time-occupation rate and the difficulties for the household to complete their tasks.

Despite the predictable completion rate drop, one may observe an interesting phenomena in relation with the housework tasks. During the first 4 weeks, Mary perform more than 70% of them because she has more spare time than John as shown on the figure 4(b). Later on, her workload rises thus, a significant part of the housework migrates toward John : Mary having less time to deal with them, those tasks are more likely to be made by John. As we set her working hours longer than his, John then takes the bigger part of the housework, around 60%. Bill, on his side, can participate a bit to the collective tasks making breakfast and lunch, the only housework tasks possible for him.

In the real world, a balanced housework sharing is not a general feature of a household. Anyway, we believe the workload allocation ability of our model to be useful as a general trend and, when necessary, several possibilities to constrain the sharing are offered to the modeller.

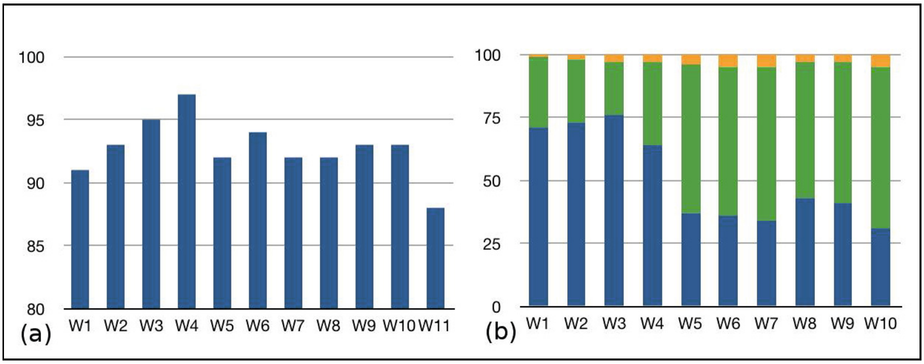


Fig. 4. (a) Completion rate in % by week and (b) Housework tasks sharing in % (Mary in blue, John in green and Bill in yellow)

5 Conclusion and Perspectives

In this paper we have presented a human behaviour meta-model and simulator. This activity-oriented meta-model enable field-experts to represent comprehensively both the routine activities of a household and the unpredictable ones(*i.e.* illness, work schedule change, etc.) that eventually happens. This meta-model is based on the quantitative definition of scheduled tasks (*i.e.* frequency and preferred periods and their degree of imperativeness). We demonstrated the capabilities of our activity model to represent both constrained and flexible habits. Moreover, two continuous processes modify such habits to cope with unforeseen events such as energy consumption incentive and a major constraint change. These features have been demonstrated on several long-term household simulations during which the electricity tariff is dynamic and the work schedule of an occupant is dramatically changed.

Our perspectives are two-fold: (1) better calibration of the current framework and (2) integration of the thermal environment. Our ESCO partner has an ongoing large scale field experiment in which precise household activity schedule are collected. We will use these to calibrate more precisely our model in terms of electricity consumption. In conjunction with participatory simulation experiments (see [10] for previous experiments), it will allow us to validate some of the hypotheses made by field-experts during the development of SMACH. Concurrently, the activity model which allows to simulation end-use consumption will be complemented with a heating model. Indeed, heating may contribute to 50% of household consumption. The under-development model is constituted of two sub-models: a thermodynamic dwelling one introduced in [7] and a thermal environment control model based on the thermal comfort as proposed by Fanger [4]. They allow to represent the thermodynamic of the dwelling, based on both its characteristic and the weather, and the inhabitants responses to the internal environmental, by taking into account the temperature but also the individuals state (*e.g.* clothing, activity, etc.).

References

1. Amouroux, E., Huraux, T., Sempe, F., Sabouret, N., Haradji, Y.: Simulating Human Activities to Investigate Household Energy Consumption. In: Proc. of ICAART (to appear 2013)
2. European Environment Agency: Energy and environment report. Tech. Rep. 6, European Commission (2008)
3. Haradji, Y., Poizat, G., Sempé, F.: Human Activity and Social Simulation, pp. 416–425. CRC Press (August 29, 2012), `http://dx.doi.org/10.1201/b12319-51`
4. van Hoof, J.: Forty years of Fanger's model of thermal comfort: comfort for all? Indoor Air 18(3), 182–201 (2008)
5. Kashif, A., Ploix, S., Dugdale, J., Le, X.H.B.: Simulating the dynamics of occupant behaviour for power management in residential buildings. Energy and Buildings (2012) (online pre-print)

6. Molderink, A., Bakker, V., Bosman, M.G.C., Hurink, J., Smit, G.J.M.: Management and control of domestic smart grid technology. IEEE Trans. Smart Grid, 109–119 (2010)
7. Plessis, G., Filfli, S., Muresan, C., Bouia, H.: Using design of experiments methods to develop low energy building model under modelica. In: IBPSA 2011 (2012)
8. Rogers, A., Maleki, S., Ghosh, S., Jennings, N. R.: Adaptive home heating control through gaussian process prediction and mathematical programming. In: Second Int. Workshop on Agent Technology for Energy Systems (ATES 2011), pp. 71–78 (May 2011)
9. Searle, J.: Speech Acts. Cambridge University Press (1969)
10. Sempé, F., Gil-Quijano, J.: Incremental and situated modeling for multi-agent based simulations. In: Proc. 2010 IEEE RIVF Conference, pp. 1–6 (2010)
11. Sierhuis, M., Clancey, W.J., Van Hoof, R.J.J.: Brahms: a multi-agent modelling environment for simulating work processes and practices. Int. J. of Sim. and Proc. Modelling 3(3), 134–152 (2007)
12. Swan, L.G., Ugursal, V.I.: Modeling of end-use energy consumption in the residential sector: A review of modeling techniques. Renewable and Sustainable Energy Reviews 13(8), 1819–1835 (2009)
13. US Department of Energy: Benefits of demand response in electricity markets and recommendations for achieving them. Tech. rep. (2006)
14. US Energy Information: Annual Energy Review 2010, US Government Printing Office (October 2011)
15. Zhou, Z., Chan, W.K.V., Chow, J.H.: Agent-based simulation of electricity markets: a survey of tools. Artificial Intelligence Review 28(4), 305–342 (2007)

MAKKSim: MAS-Based Crowd Simulations for Designer's Decision Support

Luca Crociani, Lorenza Manenti, and Giuseppe Vizzari

CSAI - Complex Systems & Artificial Intelligence Research Center
University of Milano-Bicocca
Viale Sarca 336, 20126, Milano, Italy
luca.crociani@unimib.it,{manenti,vizzari}@disco.unimib.it

Abstract. This paper presents MAKKSim, a pedestrian dynamics simulator based on a computational discrete model in which pedestrians are represented as utility-based agents. The computational model and the system architecture are discussed, focusing on the development of the tool and on its application in a real-word case study, for the comparison and the evaluation of different strategies of crowd management and of different structural changes on the geometry of the environment.

Keywords: Multi-Agent Systems, Complex Systems, Crowd, Groups.

1 Introduction

The Multi-Agent Systems approach to the modelling and simulation of complex systems has been applied in very different contexts, ranging from the study of social systems [9], to biological systems (see, e.g., [17]), and it is considered as one of the most successful perspectives of agent-based computing [13], even if it is still relatively young, compared, for instance, to analytical equation-based modelling. Models for the simulation of pedestrian dynamics and crowds have already been successfully applied to several scenarios and case studies, off-the-shelf simulators can be found on the market and they are commonly employed by end-user and consultancy companies[1]. Most of these models employ an agent-based approach, even if the notion of agent does not always relate to the one currently adopted in the autonomous agents and multi-agent systems research area: in fact, several models represent individuals, not aggregate quantities, but they interpret pedestrians as particles subject to attractive or repulsive forces. Despite the substantial amount of results this area is still quite lively: one of the least studied and understood aspects of crowds of pedestrians is represented by the implications of the presence of groups [6]. Even if recent works, especially in the agents technology area [3], represent a promising line of research, current results still need a more thorough validation and an analysis of the feasibility of their application to concrete case studies. The aim of this work is to introduce

[1] see http://www.evacmod.net/?q=node/5 for a significant although not necessarily comprehensive list of simulation platforms.

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 25–36, 2013.

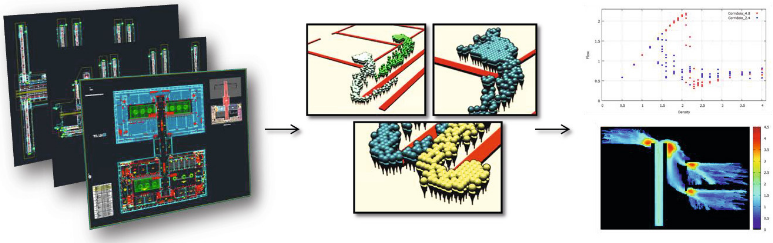


Fig. 1. The application scenario of MAKKSim: evaluation of designs and crowd management solutions by means of statistical data from simulations

MAKKSim, an innovative pedestrian and crowd simulation system considering groups as a central element of the represented scenario. MAKKSim represents an evolutionary step in the direction of an extension of robust currently employed pedestrian models towards more sophisticated behavioural models. In particular, this paper presents a version of the model including an adaptive mechanism for the preservation of group cohesion that represent a significant improvement over the previous versions [2].

The work is organised as follows: first of all, the agent-based model will be described, introducing the elements for the definition of environment, pedestrians and mechanism of movement. Then, the analysis of MAKKSim from an architectural point of view will be described (Sect. 3). The application of the platform to a real-word case study will be discussed with reference to the comparison and the evaluation of different crowd management strategies and alternative geometries of the environment for the scenario (Sec. 4). Conclusions and future developments will end the paper.

2 Agent-Based Computational Model

The agent-based computational model underlying MAKKSim will now be briefly introduced, focusing on the definition of *environment*, *pedestrians* and *movement mechanism*. A more thorough formal definition can be found in [18].

2.1 Environment

The environment is modelled in a discrete way by representing it as a grid of squared cells with 40 cm side (according to the average area occupied by a pedestrian [20]). Cells have a state indicating the fact that they are vacant, occupied by an obstacle or by a pedestrian.

A simulation scenario encompasses both the structure of the environment and all the information required for the realisation of a specific simulation, such as the demands (pedestrians generation profile, origin-destination matrices) and spatial constraints (e.g. crowd management policies). The information related to the scenario of the simulation are represented by means of *spatial markers*, special sets of cells that describe relevant elements in the environment. In particular, three kinds of spatial markers are defined: (i) *start* areas, that indicate the generation point of agents[2] in the scenario, all at once or according to a user defined *frequency*; (ii) *destination* areas, that define the possible targets of the pedestrians in the environment; (iii) *obstacles*, that identify all the non-walkable areas as walls and zones where pedestrians can not enter.

Space annotation allows the definition of virtual grids containing information about agents' positions and movements in the environment. In our model, we adopt the *floor field* approach [4], that is based on the generation of a set of superimposed grids (similar to the grid of the environment) starting from the information derived from spatial markers. Floor field values are spread on the grid as a gradient used to support pedestrians in the navigation of the environment: path fields show the shortest path towards a given destination, density fields highlight instead crowded points of the environment, and so on. Some of the floor fields are *static* (creating at the beginning and not changing during the simulation) or *dynamic* (changing during the simulation). Three floor fields are considered in the model: (i) *path field* assigned to each destination area, that indicates for every cell the distance from the destination, acting as a potential field that drive pedestrians towards it (static); (ii) *obstacles field*, that indicates for every cell the distance from an obstacle or a wall (static);(iii) *density field*, that indicates for each cell the pedestrian density in the surroundings at the current time-step (dynamic).

Chessboard metric with $\sqrt{2}$ variation over corners [11] is used to produce the spreading of the information in the path and obstacle fields; differently, for the density field, when a pedestrian p moves in a cell c, it is modified adding 1 to the cell in which he/she moves, and subtracting 1 from the cell he/she just left. For the neighbour cells, the value v decreases with the inverse of the square of the distance d between the cell and p ($v = \frac{1}{d^2}$).

2.2 Time and Update Mechanism

In our model, time is also discrete, divided into steps of intervals equal to 1/3 seconds. These configurations, along with the adoption of a Moore neighbourhood with radius equal to 1, generates a linear pedestrian speed of 1.2 ms^{-1}, in line with the data from the literature representing observations of crowd in normal conditions [20].

Regarding the update mechanism, three different strategies are usually used in this context [10]: *sequential*, *shuffled sequential* and *parallel* update. The first

[2] Different types of agents with different features can be generated by the same start area.

strategy is based on a sequential update according to a static list of priority, that reflects the generation order of the agents; on the contrary, the parallel update calculates the choice of movement of all the pedestrians at the same time. We currently adopt the shuffled sequential strategy, in which a dynamic list of priority among agents is randomly generated every step and then used to sequentially update agents positions.

2.3 Pedestrians and Movement

As we previously said, pedestrians are defined as utility-based agents:

$$Ped : \langle Id,\ Group,\ State \rangle; \quad State : \langle position,\ oldDir,\ Dest \rangle$$

with their own numerical identifier, their group (if any) and their internal state, that defines the current position of the agent, the previous movement and the final destination, associated to the relative path field.

Agent life-cycle is divided in four steps: *perception, utility calculation, action choice* and *movement*. The *perception* step provides to the agent all the information (i.e., values extracted from floor fields) he/she needs for choosing his/her destination cell. This choice is based on an utility value assigned to every possible movement according to the following function:

$$U(c) = \frac{\kappa_g G(c) + \kappa_{ob} Ob(c) + \kappa_s S(c) + \kappa_c C(c) + \kappa_i I(c) + \kappa_d D(c) + \kappa_{ov} Ov(c)}{d}$$

Function $U(c)$ takes into account all the behavioural elements relevant for pedestrian movement, combining information derived by local floor fields: *goal attraction* (i), *geometric* (ii) and *social* repulsion (iii) are the first factors considered in modelling pedestrian behaviour. In addiction, we introduce two different components to model social relationships between pedestrians: *simple group* (iv), that indicates a family, friends, or any other group characterised by a strong cohesion; *structured group* (v), generally a large one (e.g. group of team supporters), that shows a slight cohesion and a natural fragmentation into subgroups, in which the cohesion gets stronger. Moreover, two factors represent preferences with respect to movement, helping the model to reproduce realistic simulation both

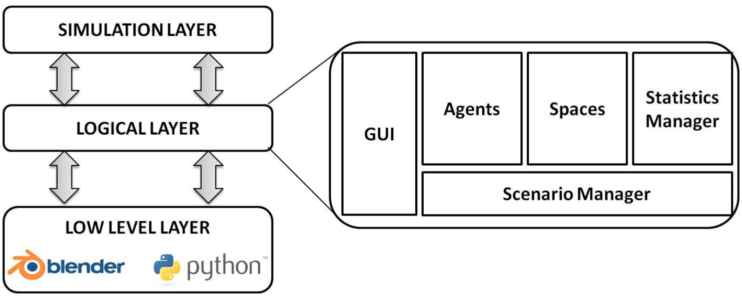
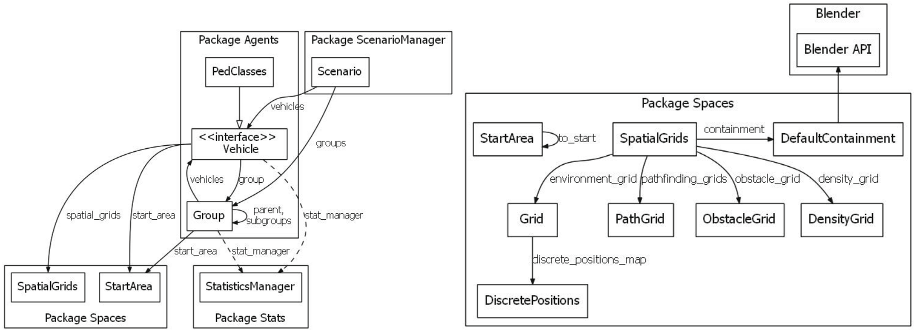


Fig. 2. High-level perspective of MAKKSim architecture



(a) (b)

Fig. 3. The class diagram of MAKKSim, from the perspective of *Agents (a)* and *Spaces (b)* packages. Note that in *(b)* links from external packages have been omitted.

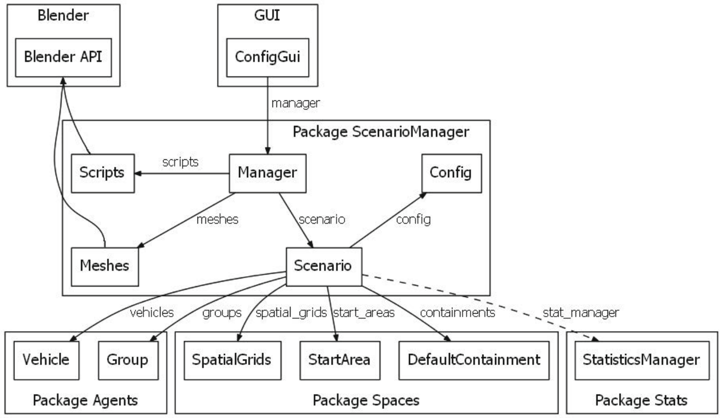
in qualitative and quantitative perspective: (vi) adds a bonus to the utility of the cell next to the agent according to his/her previous direction, while (vii) describes the *overlapping* mechanism, a method used to allow our model the possibility to treat high density situations, allowing two pedestrians occupying temporarily the same cell at the same step. The model includes an adaptive mechanism, modulating the constants that regulate the relative importance of the behavioural components, in particular to preserve the cohesion of simple groups whenever their dispersion grows above a given threshold. More details about this mechanism can be found in [18].

After the utility evaluation for all the cells in the neighbourhood, the choice of action is stochastic, with the probability to move in each cell c as (N is the normalisation factor): $P(c) = N \cdot e^{U(c)}$. The set of possible actions is defined as the list of the eight possible movements in the Moore neighbourhood, plus the action to keep the position (indicated as X): $A = \{NW, N, NE, W, X, E, SW, S, SE\}$.

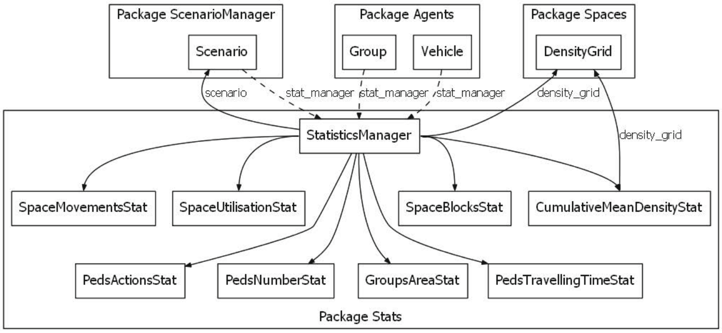
3 System Architecture

Figure 2 shows the high-level architecture of our platform. Users interact with the software through the *simulation* layer, which represents the 3D model of the environment, built using Blender[3], and the information of the simulation scenario, set by MAKKSim graphic interface. The *logical* layer contains the Python code of the simulator, structured in 5 packages: the *GUI* package (i), the *Agents* (ii) and *Spaces* packages (iii), that constitute the implementation of the computational model previously described, the *Scenario Manager* (iv), which connects the packages to run the simulation and the *Statistics Manager* (v), for statistics purposes. The last layer is the *low level* layer, that contains both the

[3] Blender is a free and open-source 3D computer graphics software, `http://www.blender.org`



(a)



(b)

Fig. 4. The class diagram of MAKKSim, from the perspective of *ScenarioManager (a)* and *Stats (b)* packages

Python virtual machine, which runs the code, and the Blender 3D engine, that supports an easy-to-read visualisation of the results of the simulations.

A more detailed explanation of the main software packages will be now presented, by means of the UML-like class diagram showed in Figure 3 and 4.

***Agents* Package.** The package *Agents* represents the portion of the software that implements the pedestrians. In order to allow the possibility to define heterogeneous types of pedestrians, the basic schema of an agent has been implemented starting from the interface *Vehicle*, in which there are four abstract methods that exactly represent the agent life-cycle: *perceive*, *evaluate*, *choose* and *update*. With this methodology, to develop a new kind of pedestrian, it is just necessary to create a new class that implements the *vehicle* interface - or extends one of the already-defined pedestrian class - and override these methods. For the representation of the element *group* class with the same name has been defined. In case of simple group, it will contain only a set of agents, otherwise it will be composed of the list of sub-groups.

***Spaces* Package.** To link the definition of pedestrians with the environment of the simulation, every pedestrian instance has the references to the main elements of the package *Spaces*: *SpatialGrids*, which represents the container of the spatial grids (including the floor fields); the relative *Start* marker of its generation. Considering the management of the environment and spatial markers, floor fields are defined respectively by the classes *PathGrid*, *ObstacleGrid* and *DensityGrid*, while the grid of the discrete environment is implemented by *Grid*, and *DiscretePositions* aggregates the positions of all pedestrians. The marker *Start* has been implemented as a class due to its complexity: it defines the generation frequency of pedestrians and groups, the type of pedestrians, the stochastic assignment of goal to agents and, finally, the reference to another *start area* to indicate where the agents will be regenerated after the goal reaching (the latter is useful to realise agents with multiple tasks or subjected to vertical movements). The interaction between MAKKSim and the Blender engine is managed by the class *DefaultContainment* that acts as a "mediator" between the continuous environment of Blender and the discrete one of MAKKSim, for the creation of the spatial grids.

Scenario Manager. In order to perform simulations, the main elements of the two previous packages are initialized and connected by the *ScenarioManager*, specifically by the object *Scenario* (Figure 4(a)). The latter is controlled by *Manager*, which is the root class of the software and runs the simulations by initialising and updating *Scenario*. Its initialisation requires the object *Configuration*, created by the GUI class with the parameters of the simulation, defined by the user. At each step the class *Scenario* is therefore updated, both with the introduction of further agents in the simulation (according to the configured frequencies) and the update of those that are already in the scenario. The elements *Meshes* and *Scripts* implement integration points with Blender: the first one is used to draw simple 3D model of the pedestrians, while the other one integrates the Python code of MAKKSim in Blender.

Statistics Manager. Class *Scenario* initialises also the core of the package *Stats*, named *StatisticsManager*, that supports the definition of statistics. We chose to implement this one as a Singleton, in reference to the well-known software architecture design patterns [8], due to the fact that it is an object which must be reachable from all agents of the simulation (and the Singleton reference can be obtained, instead of saved in each of them[4]) and, moreover, it is strictly required that only one instance of it is created.

Therefore the *StatisticsManager* works as the mediator between the agents and the statistics management objects, for the realisation of the statistical data. Classes into *StatisticsManager* control have been implemented to perform, for every step, different kind of analysis:

[4] It is the reason why in the Figure 3 and 4 the links to *StatisticsManager* are dashed.

• *SpaceUtilizationStat* saves the absolute values of cell utilisation during all the simulation, using a special discrete grid in which 1 is added in every occupied cell. It allows to analyse the average trajectories of pedestrians;
• *SpaceMovementsStat* and *SpaceBlocksStat*, similar to the previous one, focusing on the pedestrian real movements and on block situations, respectively, by adding 1 in the corresponding cells;
• *CumulativeMeanDensityStat* calculates the grid of cumulative mean density (CMD) of pedestrians[5]. This grid is obtained as follows: agents store the value of the *density field* in its position in the grid and at the end of the simulation the mean value is then calculated. Note that, with this method, CMD values of empty cells gets no variation, therefore the grid does not vary on time;
• *PedsNumberStat* saves the number of agents in the simulation;
• *PedsActionsStat* stores every action performed by the agents during all the simulation, to analyse their trajectories;
• *PedsTravellingTimeStat* stores the number of steps required by agents to reach their goal;
• *GroupsAreaStat* stores the values of pedestrian groups dispersion during the simulation. Group dispersion is a fuzzy concept and different methods can be defined to calculate it [1]. Our method identifies the dispersion of a group as the area of the convex hull generated using the position of its members.

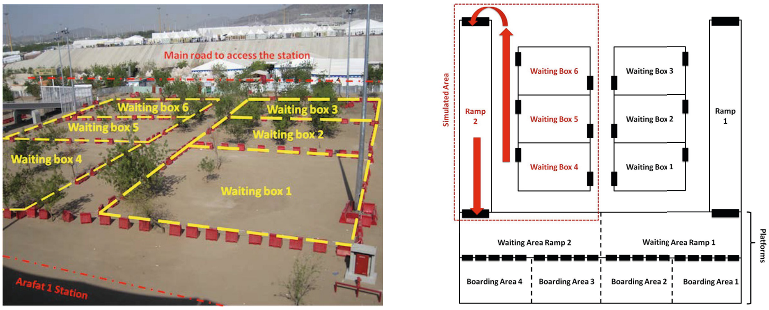
Results of the analyses are saved both as comma separated values (CSV) files (for further elaboration into a spreadsheet or plotting through tools like GNU-Plot) and as portable network graphics (PNG) images, when possible and relevant, for an immediate visual feedback.

4 Application Scenarios and First Results

Before describing and analysing the results, however, we will introduce some considerations about the computational costs of the simulations, as well as about model calibration and validation. Simulations have been performed using a workstation with an Intel(R) Xeon(R) E5630 CPU (12 MB SmartCache, 2.53 GHz clock), 6 GB Ram and Windows 7 Professional 64 bit as OS. Tests have shown up the significant computational complexity of the agent behavioural model: simulations performed in the Arafat Station scenario, that will be described later on, include a population of 750 agents, divided into three equal structured groups, and they required an average time of 14 seconds per simulation step[6].

[5] CMD is defined as the average local pedestrian density perceived from each person in each of his/her positions (see [5] for a detailed explanation).

[6] It is clear that computational time is a significant aspect with respect to the use of simulation platforms, but, during the design of the tool, it was not our aim to deal with real time simulations, that however cannot be achieved even with commercial systems but only with prototype research tools [19] and in relatively simple situations, from the point of view of pedestrian behaviour complexity. Despite that, works on optimization and parallelization of different tasks, like the statistics calculation, can be developed in the future.



(a) (b)

Fig. 5. Picture of the waiting boxes taken from the Arafat station (a) and the logical representation of the simulation scenario (b)

Regarding model calibration and validation, a set of preliminary tests on benchmark scenarios[7] have been performed, with the aim of understanding the platform capability to reproduce the aggregated macroscopic effects of the pedestrian dynamics [15]. By comparing the simulated data with the empirical evidences, we have obtained ranges of values for all the parameters in $U(c)$ function (see Sec. 2.3), allowing to perform realistic pedestrian behaviours and simulations; more details on these experimental scenarios can be found in [18]. We want to emphasise that it is this aim of producing results that are comparable to real data in a wide range of situations that calls for a relatively simple model: much richer behavioural models are present in the literature (see, e.e., [16]), but they are mostly aimed at producing visually realistic animations of environments.

The first real world application of MAKKSim was in the context of CRYSTALS[8] project: several tests have been performed on the Arafat I station on the Mashaer Metro line in Saudi Arabia to compare and analyse various solutions of crowd management, in a situation that involves large amount of pedestrians. In fact, the station is used during the Pilgrimage towards Mecca, moving a large number of people from different holy sites in a strict period of time. This scenario focuses on the entrance to the station (Figure 5). In 2010 the crowd management procedure applied in the scenario was the follows: pilgrims were divided into groups with 250 members, which were firstly directed in special zones called *waiting boxes*. The access to the ramp from waiting boxes was therefore allowed one group at a time, in order to maintain a comfortable pedestrian flow and an acceptable density on the ramp and inside the station. With the use of MAKKSim we have explored particular worst cases, in order to evaluate potential changes in the crowd management procedure, or to the environmental structure of the station.

[7] These tests had the goal to evaluate, both in quantitative and qualitative way, the model effectiveness to reproduce well-known crowd behaviours, by a set of simulations performed in simple environments, like corridors or junctions, due to the presence of sufficient empirical data available [12].

[8] `http://www.csai.disco.unimib.it/CSAI/CRYSTALS/`

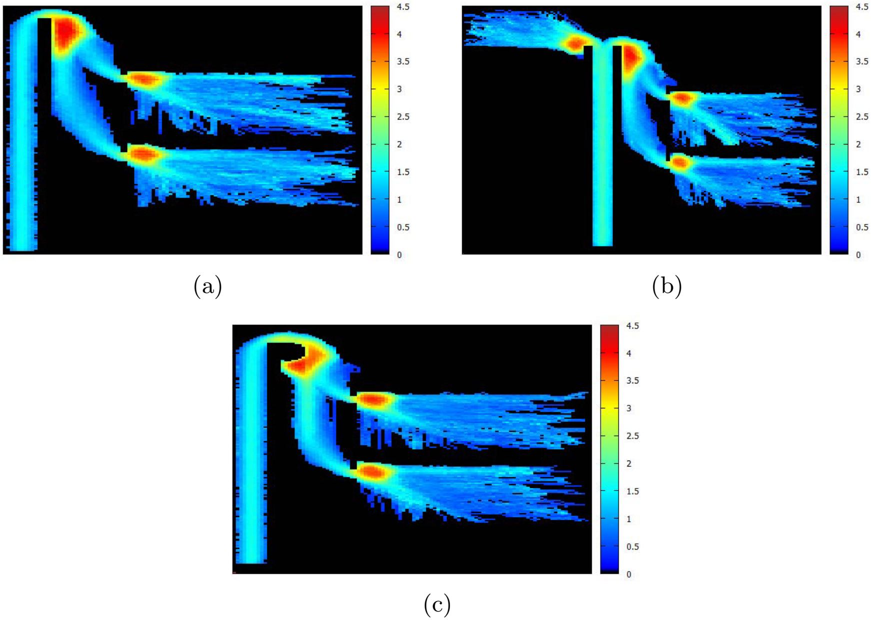


Fig. 6. CMD maps of the simulations on Arafat I station, in the three different scenarios

With the aim to restrict simulation time, we simplify the scenario by focusing on the inflow of 3 waiting boxes to the station and assuming that these ones already host one group. Every waiting box overlaps with a start area, which generates a block of 250 pilgrims subdivided in several groups, with the goal to reach the station passing through the ramp (identified as *simulated area* in Fig. 5 (b)). Starting from this configuration, three scenarios have been defined, simulated and analysed: (i) the case in which *two waiting boxes* are activated simultaneously, to verify if the parallelisation of the use of the ramp by two groups is plausible, taking into account both the ingress time and the comfort of the pilgrims; (ii) a similar situation in which an additional *external group of 250 pilgrims*, generated on the other side of the ramp, joins the groups from the waiting boxes to simulate the case in which the crowd management procedures are not respected; (iii) the case of simultaneous flow from two waiting boxes, with a *change in the geometry* of the station, introducing a large obstacle at the entrance of the ramp.

Some of the statistical results of the simulations are shown in Figure 6, representing maps of the CMD of the three scenarios. In Figure 6(a), it is possible to note that the simultaneous activation of two waiting boxes lead to a congestion near the entrance of the ramp, with a very high perceived pedestrian density (the maximum value of 4.5 m^{-2} has been reached). The total time of the simulation is improved of 10% respect with the sequential used of waiting boxes; despite that, taking into account the comfort variable, the density reached is considered not acceptable for the pedestrian walkways standard [7], so this solution should not be applied in this context. About the total time of ingress, the result of the second scenario is similar to the previous one. The problem of high densities still characterises this situation, that represents the worst case of our simulations: in Fig. 6(b) the external flow increases the density inside the ramp, that reaches a value near to 2.5 m^{-2}, also considered not acceptable. In the third scenario we try to improve the congestion situation near the entrance of the ramp, generated by the simultaneous activation of the flow from two waiting boxes: as suggested in [14] we tested the introduction of an obstacle near the most crowded point to see if this could cause a smoothening of the flow. Fig. 6 (c) shows that the round obstacle modelled leads to lower values of CMD near the entrance of the ramp. Furthermore, the average time of the simulation is decreased by 3%, because the obstacle has helped the pedestrians to better distribute themselves on the space, improving the flow.

5 Conclusions and Future Works

The paper has introduced MAKKSim, an innovative agent-based pedestrian and crowd simulation system including groups as a central element influencing system dynamics. The paper has briefly introduced the model on which the system is based, then it has shown the overall system architecture and finally a real world application based on MAKKSim has been discussed. Future works are aimed at further validating the model in additional experimental and real world scenarios, especially the group cohesion component of pedestrian's behavioural model; in addition, we intend to extend the model and the system to improve its practical applicability in more complex scenarios, modelling additional phenomenologies and practical environmental infrastructures. Finally, additional work must be done to improve the efficiency of the simulation, in particular through the potential parallelisation of some tasks.

References

1. Bandini, S., Rubagotti, F., Vizzari, G., Shimura, K.: An agent model of pedestrian and group dynamics: Experiments on group cohesion. In: Pirrone, R., Sorbello, F. (eds.) AI*IA 2011. LNCS, vol. 6934, pp. 104–116. Springer, Heidelberg (2011)
2. Bonomi, A., Manenti, L., Manzoni, S., Vizzari, G.: Makksim: Dealing with pedestrian groups in mas-based crowd simulation. In: WOA. CEUR Workshop Proceedings, vol. 741, pp. 166–170. CEUR-WS.org (2011)
3. Bosse, T., Hoogendoorn, M., Klein, M.C.A., Treur, J., van der Wal, C.N., van Wissen, A.: Modelling collective decision making in groups and crowds: Integrating social contagion and interacting emotions, beliefs and intentions. Autonomous Agents and Multi-Agent Systems 27(1), 52–84 (2013)
4. Burstedde, C., Klauck, K., Schadschneider, A., Zittartz, J.: Simulation of pedestrian dynamics using a two-dimensional cellular automaton. Physica A: Statistical Mechanics and its Applications 295(3-4), 507–525 (2001)
5. Castle, C., Waterson, N., Pellissier, E., Bail, S.: A comparison of grid-based and continuous space pedestrian modelling software: Analysis of two uk train stations. In: Pedestrian and Evacuation Dynamics, pp. 433–446. Springer US (2011)

6. Challenger, R., Clegg, C.W., Robinson, M.A.: Understanding crowd behaviours: Supporting evidence. Tech. rep., University of Leeds (2009)
7. Fruin, J.J.: Pedestrian Planning and Design. Metropolitan Association of Urban Designers and Environmental Planners, New York (1971)
8. Gamma, E., Vlissides, J., Helm, R., Johnson, R.: Design patterns: Elements of reusable object-oriented software. Addison - Wesley (1995)
9. Gilbert, N., Troitzsch, K.G.: Simulation for the Social Scientist, 2nd edn. Open University Press (2005)
10. Klüpfel, H.: A Cellular Automaton Model for Crowd Movement and Egress Simulation. Ph.D. thesis, University Duisburg-Essen (2003)
11. Kretz, T., Bönisch, C., Vortisch, P.: Comparison of various methods for the calculation of the distance potential field. In: Pedestrian and Evacuation Dynamics 2008, pp. 335–346. Springer, Heidelberg (2010)
12. Kretz, T., Grünebohm, A., Kaufman, M., Mazur, F., Schreckenberg, M.: Experimental study of pedestrian counterflow in a corridor. Journal of Statistical Mechanics: Theory and Experiment 2006(10), P10001 (2006)
13. Luck, M., McBurney, P., Sheory, O., Willmott, S. (eds.): Agent Technology: Computing as Interaction. University of Southampton (2005)
14. Nishinari, K., Suma, Y., Yanagisawa, D., Tomoeda, A., Kimura, A., Nishi, R.: Toward Smooth Movement of Crowds. In: Pedestrian and Evacuation Dynamics 2008, pp. 293–308. Springer, Heidelberg (2010)
15. Schadschneider, A., Klingsch, W., Kluepfel, H., Kretz, T., Rogsch, C., Seyfried, A.: Evacuation Dynamics: Empirical Results, Modeling and Applications. ArXiv e-prints (2008)
16. Shao, W., Terzopoulos, D.: Autonomous pedestrians. Graphical Models 69(5-6), 246–274 (2007)
17. Textor, J., Hansen, B.: Hybrid simulation algorithms for an agent-based model of the immune response. Cybernetics and Systems 40(5), 390–417 (2009)
18. Vizzari, G., Manenti, L., Crociani, L.: Adaptive pedestrian behaviour for the preservation of group cohesion. Complex Adaptive Systems Modeling (in press 2013)
19. Wagoum, A.U.K., Chraibi, M., Mehlich, J., Seyfried, A., Schadschneider, A.: Efficient and validated simulation of crowds for an evacuation assistant. Journal of Visualization and Computer Animation 23(1), 3–15 (2012)
20. Weidmann, U.: Transporttechnik der fussgãnger: transporttechnische eigenschaften des fussgãngerverkehrs (literaturauswertung) (1993), http://infoscience.epfl.ch/record/41377

A Multiagent System for Resource Distribution into a Cloud Computing Environment

Fernando De la Prieta[1], Sara Rodríguez[1], Javier Bajo[2], and Juan Manuel Corchado[1]

[1] Department of Computer Science and Automation Control, University of Salamanca, Plaza de la Merced s/n, 37007, Salamanca (Spain)
[2] Department of Artificial Intelligence, Technical University of Madrid, 28660, Campus Montegancedo, Boadilla del Monte, Madrid (Spain)
{fer,srg,corchado}@usal.es, jbajo@fi.upm.es

Abstract. It is undeniable that the term Cloud Computing has gained in importance at a remarkable pace. It is a technology which is becoming a common element of our life, due to the variety of devices related to the Internet of Things. In this technological frame, there are not many studies in which a Multiagent system has facilitated the management of a cloud-based computational environment; although a first sight its features (autonomy, decentralization, auto-organization, etc.) seem suitable for the task. This study presents the +Cloud which is a cloud platform managed by a Multiagent System.

Keywords: Cloud Computing, Multi-agent System. Allocating resources.

1 Introduction

Cloud computing is being positioned as one of the main technological paradigms to have appeared in recent years. Among its most notable features [14] is its capability to offer any type of computational services; in addition to a pool of variable resources that satisfy the computational needs of the offered services. This means that it is possible to provide new resources to the services in an elastic way according to current demand.

The reasons for the quick growth of the computational paradigm are varied, but it is possible to group them into three main categories. The first group is formed by the main technology companies (IBM, Google, Amazon, Microsoft, etc.) who have an economic interest in this paradigm as a new market. While previously an emergent market, its current dominance can be explained, in part, by a new business model which does not require an initial investment; instead, the client simply pays according to the resources used (*pay-as-you-go* [4]) These companies oriented their efforts (economic, technological and human) to the development of this technology by creating various pilot projects (Sun Cloud by Sun, Blue Cloud by IBM, etc.), in addition to other open approaches [21] which eventually resulted in what we know as Cloud Computing. Secondly, the quick birth has been possible as a result of the maturity of the variety of technological components (server, cluster, high availability, grid computing) that form the computational paradigm; as well as the tremendous research, at

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 37–48, 2013.

both the hardware and software level, in incipient technologies such as virtualization [3]. Finally, there cannot be the slightest doubt that positive public reception has been a key factor in its rapid development. From the public's perspective, a cloud environment makes it possible not only to synchronize data, information, or even tasks, projects, etc., but also to work in a delocalized way through the use of online tools. And, from the companies' perspective, the main advantage is that a cloud environment does not require an initial investment, making it possible to pay only for those resources that are required at a particular moment.

Multi-agent Systems (MAS) have not played an important role in the development of Cloud technology, as Talia presents in this study [22]. According to Talia, it is possible to distinguish to groups: (i) MAS that use the computational features of a Cloud environment (processing, storage, etc.); and (ii) Cloud environments that use MAS for the internal management of their resources, or to offer intelligent services. As is shown in the following section, the state-of-the-art indicates that the majority of the applications are related to the former group (agents using Clouds) [12][17]. Within the latter group, however, it is only possible to find incipient research lines focused on applying agreement and negotiation technologies to the task of allocating the internal resources of the cloud [1], or reaching an agreement on the quality of the cloud services being offered [15].

Although there is no clear convergence between MAS and Cloud Computing, it is obvious that the MAS features (dynamicity, flexibility, autonomy, proactivity, learning, etc.) are exactly the features that a Cloud environment needs for the self-management of its resources. A Cloud Computing environment has a set of resources (physical and virtual) which have to vary dynamically in order to cope with the demand of the computational services being offered. Within this model, the decision making process is complex, due to the variability of the demand for services and the lack of information on the decision components. This is the why an agent-based Cloud computing environment is suitable for the efficient allocation of computational resources, enabling the dynamic and automatic readaptation of each element which forms part of the cloud environment.

This study presents the +Cloud (masCloud) platform that offers Cloud services such as storage of information and files at the platform level, as well as a set of native software applications. The core of this platform is a MAS based on a Virtual Organization (VO), which makes it possible to automatically manage the computational resources of the system, adapting them in an elastic way according to demand.

The present paper is structured as follows: the following section presents the state-of-the art of Cloud computing system and its relationship with MAS. The +Cloud architecture is then presented in detail. This study finalizes with a review of the initial tests of the system, conclusions and future research lines.

2 Cloud Computing and Multiagent Systems

Cloud Computing, understood as computational paradigm, is emerging recently with great importance. Although it may be initially considered another computational paradigm, reality indicates that its rapid progression is motivated by economic interests [4] in the underlying computational features.

Historically, the term *Cloud Computing* was first used by Professor Rammath. [5]. However, the concept was becoming popular through Salesforce.com, a company that focused its market strategy to offer software as a service (SaaS) to big companies. However, IBM was the first company to detail the specific terms of the guidelines of this technology (auto-configuration, auto-monitorization, auto-optimization) in the document *Autonomic Computing Manifesto* [20]. By 2007, Google, IBM and others had joined together to form a research consortium which resulted in the birth of this technology as we know it today. [13]

For the large companies, knowledge about this technology is a competitive advantage. First of all, the Cloud provider can offer its services through a pay-as-you-go model [4] [9], following the guidelines proposed by Utility computing [16]. Additionally, the Cloud user does not have to be concerned with demand peaks, transforming passive investments in operational expenses [2].

A large number of definitions [4][14][9][2] have emerged at both a company and academic level. In each one, the authors try to highlight the most relevant features from their point of view. When a wide number of definitions are analyzed, it is possible to distinguish two big groups:

1. Those whose interests are focused on defining the technological aspects of the computational paradigm; these can be further divided in those who focus on defining either hardware or software characteristics
2. Those whose interest is to highlight the aspects related to the negotiation model, which is intrinsically associated with a Cloud environment.

It was the American NIST (National Institute of Standards and Technology)[1] which defined Cloud Computing [3] as a *model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction*. Further to this definition, NIST presents a set of features, different deployment models (private, public Community Cloud); and most importantly, three models of service. Understanding service as a capability that the Cloud offers to the end users, we can underscore the following three service models:

- Software as a Service (SaaS). This capability allows the provider to supply the user with applications that can be directly executed on the cloud infrastructure. This entails a number of advantages such as the ubiquity of the applications or the use of light clients. However, there are also a number of difficulties (which in some cases are strengths) directly related to the consumer's loss of control over the infrastructure (network, storage, operating system, difficulty to configure, etc.).
- Platform as a Service (PaaS). This capacity is supplied by the provider and allows the consumer to use the necessary tools to create their own applications within the

[1] http://www.nist.gov/

Cloud environment. Some of these services include programming, libraries, tools, etc.
As with the services in the previous level, the programmer does not control the underlying infrastructure, nor the environment where the applications are deployed.

- Infrastructure as a Service (IaaS); (o Hardware as a Service for Wang *et al.* [24]). This capability provides to the consumer include different kinds of hardware such as processing, storage, network, etc.
  This capacity can be provided to consumers with the ability to install their own software in an operating system deployed in a hardware environment, obviously virtualized, with characteristics defined by the actual user.

This division of models leads some authors to speak of Something as a Service (*ssS) [22].

2.1 Cloud and Agents

As indicated in the introduction, there is only a limited number of studies in the state of the art that relate Cloud Computing and agent technology [22]. In general terms, a Cloud system may use MAS applications in a Cloud environment for deployment, and there are also Cloud environments that use agent technology to manage their resources. Some of those applications include:

- **Agents using Cloud.** Within this group, the main state of the art applications use computational resources from the Cloud environment. For example, there are systems such as those described in [7][8] that use the computational strength of the environment to perform simulations in different fields. Another example is presented in [6], where the Cloud environment is used as a persistence engine for information.
- **Cloud using Agents.** Within this subgroup, the range of possibilities is even further extended. Mong Sim [15] highlights three subgroups of applications: (i) combination of resources among Cloud providers; (ii) planning and coordination of shared resources; (iii) establishing contracts between users and Cloud service providers. As Mong Sim points out, it is possible to find studies such as [11][23] that develop a Cloud service using agents for different specific purposes. Mong Sim used the Cloudle [15] which is an agent-based tool for discovering Cloud services. Some notable examples of Cloud providers combining resources include studies by Kaur Grewal [10] and Aarti Singh [19], which use shared Cloud resources to offer Infrastructure as a Service (IaaS) Examples that apply SLA to distribute services include [15][18]. Finally we should point out the application of negotiation and agreement algorithms applied to different levels and processes within the framework of a cloud computing environment [25].

3 Proposed Architecture: +Cloud

+Cloud is a platform based on the cloud computing paradigm. This platform allows offering services at the PaaS and SaaS levels. The IaaS layer is composed of the physical environment which allows the abstraction of resources shaped as virtual machines; however, the system does not offer this kind of service to the end users as shown in Figure 1.

+Cloud has a layered structure that covers the main components of cloud computing:

- The SaaS layer is composed of the management applications for the environment (control of users, installed applications, etc.), and other more general third party applications that use the services provided by the next layer (PaaS).
- The PaaS layer provides services through REST web services in API format. One of the more notable services among the APIs is the identification of users and applications, a simple non-relational database service and a file storage area that controls versions and simulates a directory structure.
- The IaaS layer is used to deploy all management and general-purpose applications, in addition to the all services at the platform layer. This layer provides a virtual hosting service with automatic scaling and functions for balancing workload.

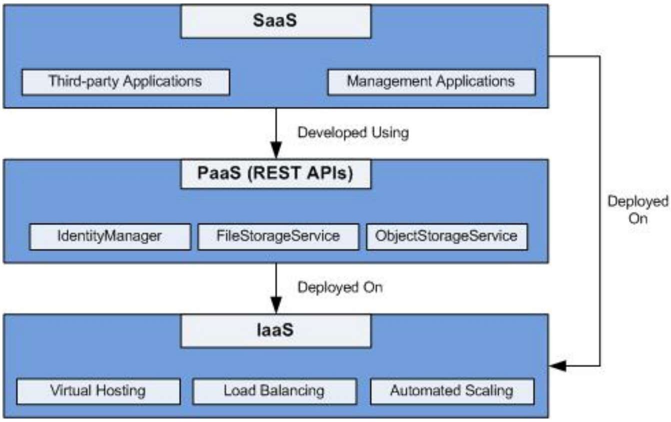


Fig. 1. +Cloud layer overview

From an internal viewpoint at a technological level, a Cloud Computing system is formed by a large set of computational resources, referred to in previous technologies as a server cluster or server farm. Abstractions are performed over these hardware resources, as virtual machines, which allows the easy and dynamic management of computational resources. Although performance decreases as a result of the computational needs of managing virtual computational resources, the advantages exceed the disadvantages, since complex tasks, such as the creation/destruction of virtual machines based on templates, the dynamic configuration of assigned resources, or even the migration of virtual machines between physical servers without stop the pending task, are made possible by virtualization.

In conclusion, a Cloud computing environment such as +Cloud platform can be viewed, at an external level, as a set of computational resources offered to end users. At an internal level, these services are deployed into a set of virtual machines that are hosted by physical server of the computational environment, as shown in Figure 2.

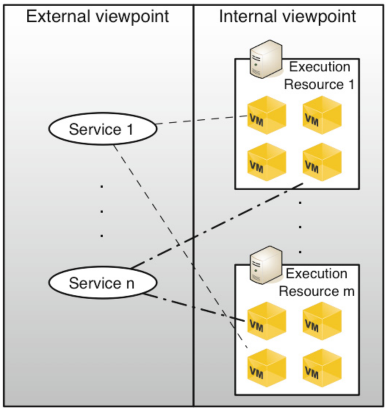


Fig. 2. Cloud Computing deployment

The distribution of physical resources between the different virtual machines and between the different system services is a matter of current interest [25][15][19]. The redistribution of resources can be seen from both a micro and macro level point of view. From a micro point of view, there is a distribution of resources between the virtual machines that accommodate a single host. In other words, a physical server has a set of physical resources available (processing, memory and drive) that must be shared among the different virtual instances that it hosts, leaving a set of minimum resources available for its own host. At the macro level, there is a redistribution of resources at a global level in the Cloud, which entails migrating virtual machines in use between different servers, and turning on and off the physical machines that provide or consume resources within the Cloud environment.

The +Cloud platform uses a virtual organization of agents to manage the system resources. MAS can be perfectly adapted to solve this problem, as it allows making decisions in an open environment where the availability of information is limited and agents are thereby required to make decisions, amidst great uncertainty, that affect the entire system. As the decision making is a distributed process, the system has greater availability than other systems in which decision making is a centralized process.

Figure 3 provides a high level description of the system. As shown, the system is divided into the following agent organizations:

- **Resource Organization.** This agent organization is charge of managing both the physical and virtual system resources. The agents are distributed throughout the hardware elements of the Cloud environment. Their main goal is to maximize the use of resources. It is intended that there are no active resources that are underutilized, which implies that there must be the smallest possible number of active physical machines to satisfy the current demand. At the same time, the computational load of the active physical resources must be high. Within this organization includes the following roles:

- *Local Manager*, in charge of allocating the resources of a single physical machine among its virtual machines and its own physical machine. There is one in each physical server.
  In terms of its internal architecture, a CBR-BDI [26] agent can redistribute the resources of each physical machine among the different virtual machines according to the partial information that it has. It can modify (increase or decrease) the resources of the physical machines in use.
  Additionally, it can start up or shut down virtual machines within the local server; it does not do so by its own initiative, however, since it is the Global Regulator agent within the same machine that gives the order.
- *Local Resource Monitor*, in charge of knowing the usage level of the virtual resources of each virtual machine. There is one monitor for each physical machine and it has all the knowledge about the physical machine as well as its virtual machines.
- *Global Regulator*, in charge of negotiating with its peers regarding the redistribution of the resources at a global level. There is one in each physical server.
  The Global Regulator uses agreement algorithms between peers to distribute resources at a global level. When the service does not have the desired quality, all agents throughout the system with this role must reach an agreement as to how to solve the problem. To do so, they will use a distributed CBR system and algorithms according to [25]. Once the decision has been made, it is applied to the system in order to solve the problem.
- *Network Monitor*, this role can monitor the network from the point of view of each single physical machine. There is one in each physical server.
- *Hardware Manager*, the goal of this role is to manage at all times the hardware that is both in use and on standby. There is one in each physical server, each of which acts as coordinator.

• **Consumer Organization.** At the technological level this organization deploys over the computational resources offered by the organization described in the previous section. The services encompassed by this organization will, therefore use the system resources according to existing demand. Its main goal is to maximize the quality of service, which requires monitoring each service individually, keeping in mind that each service will be deployed simultaneously on various virtual machines located in different physical services.
  - *Service Supervidor,* this role is responsible for making decisions about each individual service. There is one for each service, each of which is located in the same virtual machine hosting the SDM of the same service, which in turn incorporates the load balancer service.

  – *Service Demand Monitor*, in charge of monitoring each demand service which is offered by +Cloud. There is one agent of this type per each kind of service. They incorporate a load balancer to redirect requests to the different virtual machine which are offering the service at that time.
  – *User,* represents the system users that use the services. As such, they are the ones that ultimately use the system resources. There can be different types of users: SaaS User, Cloud User and Administrator.
- **Management Organization.** This organization is in charge of ensuring that the entire system functions correctly, which is in fact its primary goal. There are two types of roles:
  – *Global Supervisor,* which ensures that the other roles and agents of the VO work correctly. If something fails, or one of the agents does not respond to its messages, this role will take the necessary actions to restore the system to a functioning state.
  – *Identity Manager*, which is in charge of allowing other agents to enter or exit the system. Within the framework of this system, each time that a service is initiated or suspended, as with a physical machine, agents will enter or exit the system. This manager is also in charge of logging the User agents. As with all other services, this role, which itself constitutes a service, must be monitored as such.

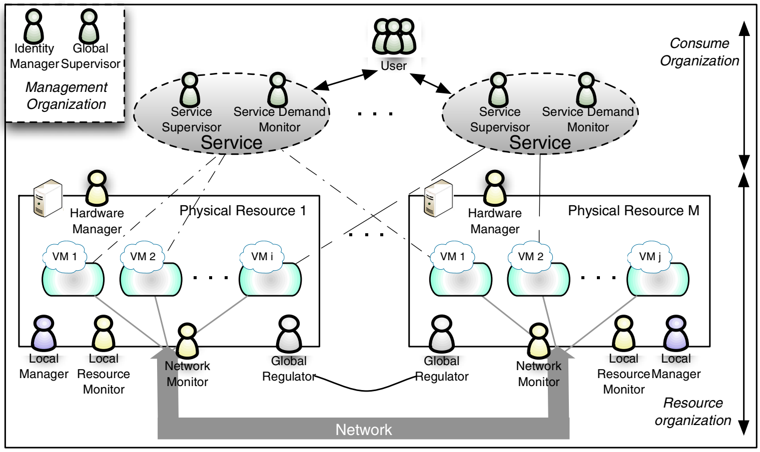


Fig. 3. MAS for resource redistribution in +Cloud

4 Preliminary Results

The +Cloud architecture has been continually evolving since its beginning. Beyond the multiagent system that governs the computational environment, it is also necessary to develop a set of services that are offered to the end user. Some of the more important services include FSS (*File Storage Service*), which stores files, and OSS (*Object Storage Service*), which stores information in a non-SQL database. The elasticity of these Cloud services is supported in the +Cloud architecture previously presented.

In order to obtain data regarding the performance of the Cloud environment and its ability to adapt to changes in the demand of services, a Denial of Service (DoS) attack is executed over the FSS. This is done by sending a constant stream of requests to the service over a period of 300 seconds. The number of requests is continually increased during the time of the test. This section presents the results obtained.

As indicated, the adaptation can be seen from both a micro and macro perspective. At the micro level, the adaptation takes place in each of the individual servers within the Cloud environment. This provides the physical server with limited capabilities (processing, memory, hard drive, etc.), which it must then share among the virtual machines it hosts. The adaptation takes place through the *Local Manager* agent, which works closely with the *Local Resource Monitor* agent. These agents are local to each machine and do not have information about the other nodes in the Cloud environment. The *Local Manager* agent uses a CBR as a support system for making decisions. The information that this agent has is provided by the monitor, which is continually gathering data about the computational load of each machine in the physical server. Figure 4 shows the adaptation of the system. The graph on the left shows an increase in the use of the processor (blue line) and the corresponding increase of CPU assigned by the Local Manager (red line). The graph on the right shows the same process, except that the memory is assigned to a virtual machine; the blue line is the memory used while the red line is the memory assigned.

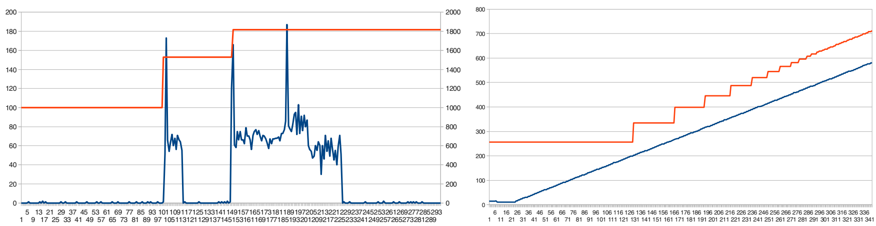


Fig. 4. Redistribution of resources at micro-level (Left: processor, Right: memory)

When the *Local Manager* detects insufficient resources, or the *Service Supervisor* detects decreased quality of service, an adaptation process takes place at a macro level. This process, which can be reviewed in detail in [25], is based on the negotiation among the Global Regulator agents for different nodes in the Cloud environment. During this negotiation process, the agents decide how to redistribute resources among the different nodes (and not just internally) to rectify the problem of demand. Figure 5 shows how quality of service is improved. While the response time (y-axis) in the first part of service is very high, the quality of service improves considerably after two consecutive readaptations.

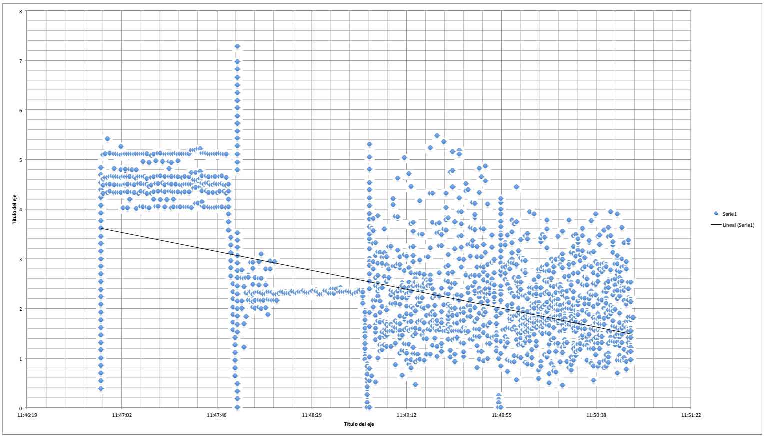


Fig. 5. Redistribution of resources at macro-level

5 Conclusions and Future Work

This study has presented the +Cloud platform, which is a Cloud Computing platform that is managed at internal level for a MAS based on a VO. As indicated in the introduction, initial results have shown that MAS technology is ideal for managing the computational resources of this type of system.

State of the art Cloud environments follow a centralized model for making decisions [21], which can lead to different problems such as (i) the need to centralize information; (ii) the need for a large computational load in the nodes where the decision making process takes place; and (iii) finally, the ability to recover from mistakes that can arise with a centralized decision making process. Using a model such as that proposed in this study, all of these problems can be resolved. This is due, in large part, to the system agents, which are able to make decisions in a Cloud environment where only partial information is available. Even if one or more nodes fail, it is still possible for the readaptation process to take place in the available nodes. Finally, the use of MAS allow using such techniques as the agreement techniques described in [25], which makes it possible to make decisions for readaptation at a global level without needing to centralize the information.

A final note with regard to future lines of work, given the great technological component of the system and its dependency on the environment, we expect that future versions of the Cloud environment will include the model proposed by Ricci in [27], which introduces the concept of Artifact to interact with the environment. The use of this model will facilitate the evolution of the platform and its independence from the underlying technological environment.

Acknowledgment. This work is supported by the Spanish government (MICINN), project OVAMAH, reference: TIN2012-36586-C03-01.

References

1. An, B., Lesser, V., Irwin, D., Zin,, M.: Automated Negotiation with Decommitment for Dynamic Resource Allocation in Cloud Computing. In: AAMAS 2010 Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems, vol. 1, pp. 981–988 (2010)
2. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: A view of Cloud Computing. Communications of the ACM 53(4), 50–58 (2010)
3. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xend and the Art of Virtualization. In: ACM Symposium on Operating System Principles, Boltoin Landing, NY, USA, pp. 164–177 (2003)
4. Buyya, R.: Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities. In: 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGRID 2009, pp. 5–13 (2009)
5. Chellappa, R.: Intermediaries in cloud-computing: A new computing paradigm. In: Informs. Cluster: Electronic Commerce, Dallas, Texas (1997)
6. Chen, C., Wang, K.: Cloud Computing for agent-based urban transportation system. IEEE Intelligent Systems 26, 73–79 (2011)
7. Cheng, Y., Low, M.Y.H., Zhou, S., Cai, W., Seng Choo, C.: Evolving agent-based simulations in the clouds. In: Third International Workshop on Advanced Computational Intelligence (IWACI), pp. 244–249 (2010)
8. Dębski, R., Byrski, A., Kisiel-Dorohinicki, M.: Towards and agent-based augmented cloud. Journal of Telecommunications and Information Technology, 16–22 (2012)
9. Erdogmus, H.: Cloud Computing: Does Nirvana Hide behind the Nebula? IEEE Sofware 26(2), 3–6 (2009)
10. Grewa, R.K., Pateriya, P.K.: A Rule-based Approach for Effective Resource Provisioning in Hybrid Cloud Environment. International Journal of Computer Science and Informatics, 101–106 (2012)
11. Jin Kim, M., Gun Yoon, H., Ku Lee, H.: IMAV: An intelligent Multi-agent model based on Cloud computing for resource virtualization. In: Computers, Networks, Systems, and Industrial Engineering, pp. 99–111 (2011)
12. Li, Z., Chen, C., Wang, K.: Cloud Computing for Agent-based Urban Transportation Systems. IEEE Intelligent System 26, 73–79 (2011)
13. Lohr, S.: Google and ibm join in cloud computing research. New York Times (2007)
14. Mell, P., And Grance, T.: The Nist Definition of Cloud Computing. In: NIST Special Publication 800-145. pp. 1–3. NIST (2011)
15. Mong Sim, K.: Agent-based Cloud Commerce. In: IEEE International Conference on Industrial Engineering and Engineering Management, pp. 717–721 (2009)
16. Ross, J.W., Westerman, G.: Preparing for utility computing: The role of IT architecture and relationship management. IBM Systems Journal 43(1), 5–19 (2004)
17. Schuldt, A., Hribernik, K.A., Gehrke, J.D., Thoben, K.D., Herzog, O.: Cloud Computing for Autonomous Control in Logistics. In: 10th Annual Conference of the German Society for Computer Science (2010)
18. Siebenhaar, M., Nguyen, T.A.B., Lampe, U., Schuller, D., Steinmetz, R.: Concurrent Negotiations in Cloud-based Systems. In: Vanmechelen, K., Altmann, J., Rana, O.F. (eds.) GECON 2011. LNCS, vol. 7150, pp. 17–31. Springer, Heidelberg (2012)

19. Singh, A., Malhotra, M.: Agent based framework for scalability in Cloud Computting. International Journal of Computer Science & Engineering Technology (IJCSET), 41–45 (2012)
20. Stoica, F., Morris, D., Karger, M., Kaashoek, F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM 2001, pp. 149–160 (2001)
21. Takato Endo, P., Estácio Gonçalves, G., Kelner, J., Sadok, D.: A survey on Open-source Cloud Computing Solutions. In: Brazilian Symposium on Computer Networks and Distributed Systems, pp. 3–16 (2010)
22. Talia, D.: Clouds Meet Agents: Towards Intelligent Cloud Services. IEEE Internet Computing 16(2), 78–81 (2012)
23. Venkataramana, K., Padmavathamma, M.: Agent-based approach for Authentication in Cloud. IRACST - International Journal of Computer Science and Information Technology & Security 2(3), 598–603 (2012)
24. Wang, L., Tao, J., Kunze, M., Castellanos, A.C., Kramer, D., Karl, W.: Scientific Cloud Computing: Early Definition and Experience. In: HPCC 2008: 10th IEEE International Conference on High Performance Computing and Communications, pp. 825–830 (2008)
25. Heras, S., De la Prieta, F., Julian, V., Rodríguez, S., Botti, V., Bajo, J., Corchado, J.M.: Agreement technologies and their use in cloud computing environments. Progress in Artificial Intelligence 1(4), 277–290 (2012)
26. Corchado, J.M., Pavón, J., Corchado, E., Castillo, L.F.: Development of CBR-BDI agents: A tourist guide application. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, pp. 547–559. Springer, Heidelberg (2004)
27. Ricci, A., Omicini, A., Viroli, M., Gardelli, L., Oliva, E.: Cognitive Stigmergy: Towards a Framework Based on Agents and Artifacts. In: Weyns, D., Van Dyke Parunak, H., Michel, F. (eds.) E4MAS 2006. LNCS (LNAI), vol. 4389, pp. 124–140. Springer, Heidelberg (2007)

Empirical Specification of Dialogue Games for an Interactive Agent

Guillaume Dubuisson Duplessis, Nathalie Chaignaud, Jean-Philippe Kotowicz, Alexandre Pauchet, and Jean-Pierre Pécuchet

LITIS - EA 4108, INSA Rouen, Avenue de l'Université - BP 8,
76801 Saint-Étienne-du-Rouvray Cedex, France
gdd@litislab.eu, {name}@insa-rouen.fr

Abstract. This article addresses the challenge of designing the communicative behaviour of an agent interacting with humans. We present a data-driven methodology based on the production of a matrix representation of a corpus from which we extract dialogue patterns. These patterns reflect the minimal units of interaction which turn out to be very attractive for dialogue modelling. We present a framework to specify dialogue games from these patterns based on the notion of social commitments. We exemplify the specification of dialogue games by implementing all the steps of our methodology on a task-oriented corpus. The produced games are validated by showing that they appropriately describe the patterns appearing in a reference corpus.

1 Introduction

Interactions between humans and software agents become commonplace in heterogeneous multiagent systems and mixed communities. However, designing agents interacting with humans is known to be a difficult task (see issues related to mixed-initiative reasoning [1]). Indeed, humans use robust communication and reasoning processes to which agents must adapt. In particular, one challenge in conceiving human-agent interaction is the *communication issue*. Dialogue management has been spotted as being a key feature [1] because it is a very efficient means of communication for people which requires little or no training to use. Besides, it is the most likely way to achieve true, mixed-initiative, collaborative systems. Nevertheless, dialogue management remains a major deadlock in Embodied Conversational Agents (ECAs) [2]. Most of them only integrate basic dialogue management processes, such as a keyword spotter within a finite-state approach or a frame-based approach.

From our perspective, the design of ECAs can be improved by analysing and modelling human-human and human-agent interactions. To this purpose, we presented a *data-driven* methodology which aims at improving the interaction capabilities of agents interacting with humans [2]. This methodology is based on the *collection* of a corpus of dialogues thanks to a user experiment designed on purpose and depending on the future agent that is being modelled. Dialogues in this corpus are turned into a *matrix representation* through an annotation step.

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 49–60, 2013.

Then, *interaction patterns* are extracted and form the basis of the *interaction model* of the agent. Considering our matrix representation, a *dialogue pattern* is defined as a sequence of annotations which arrangement occurs in several dialogues. In other words, a *dialogue pattern* is an ordered set of utterances that is frequently reoccurring during dialogues (e.g., a question/answer pair).

The main focus of this article is the modelling of dialogue patterns obtained through the methodology into dialogue games [3]. These dialogue games constitute the basic interaction units that should be next integrated into the deliberative process of an agent. To illustrate our purpose, we present the full implementation of our methodology on a corpus from collection to pattern extraction. Then we show how to model dialogue games from dialogue patterns.

Section 2 draws some links with related work, with particular attention on existing connections between dialogue patterns and dialogue management. Section 3 describes the corpus used to illustrate our approach (annotation and extraction steps). Section 4 presents the framework used to model dialogue patterns observed in the corpus and its link to dialogue management. Section 5 shows a validation of the modelled games against the reference corpus. Lastly, section 6 concludes this article.

2 Related Work

One striking observation in a human dialogue corpus is the presence of *interaction patterns* [2,3]. In dialogue modelling, this has been analysed both as being evidence of a *plan* from the interlocutors (plan-based approach) and as a manifestation of *conventional* devices used by dialogue participants (conventional approach). The first approach focuses on the *intentional structure* of the dialogue [4]. It lies on the representation and reasoning about the underlying intentions behind dialogue participant utterances. Basically, this approach considers that a speaker utterance conveys an intention that is part of a plan. Then, it is the addressee's task to infer it and respond accordingly to the underlying plan (rather than just to the speaker's utterance). This approach leads to influential results such as the TRAINS system [5] or Collagen [6]. The second approach aims at studying the *interaction patterns* without focusing on the underlying intentions. It comes from the observation that many types of utterances do not seem to be consciously emitted but rather conventionally triggered by the context (e.g., greetings). This leads to the production of rules that describe admissible sequences of utterance types. These reoccurring patterns can be studied either in terms of *dialogue grammars* [7] or *dialogue games* [3]. These two approaches are seen as opposed whereas some researchers strongly argue that they are actually *complementary* [8,9]. This is based on the fact that communication processes can be considered as joint actions between a speaker and hearers [10]. The key characteristic of a joint action is the *coordination* of *participatory actions* by two or more people. However, people cannot deliberate indefinitely in an opportunistic activity such as dialogue. Hence, coordination must stand on devices like *conventions* which are reflected by interaction patterns. Moreover, these authors have spotted some weaknesses of the plan-based approach. Namely, the plan-recognition process remains a very complicated task and is technically speaking difficult to set up [11]. Next, dialogue is an *opportunistic activity* [10]. Consequently, not only are some sentences more likely to be conventionally triggered by the situation but some sequences of acts can not be planned as well [12] (e.g., clarifications). In the light of these results, dialogue is viewed as a shared and dynamic activity that requires both high-level deliberative reasoning and low-level reactive responses.

Here, we focus on dialogue games used to explain human dialogue and to generate artificial dialogues dedicated to humans [3]. A dialogue game is a *bounded* activity with an *entry* and an *exit* where participants play a *role* (initiator or partner). *Rules* of the game specify the expected *moves* for each participant. Participants are expected to play their roles by making the moves expected by the current *stage* of the game. To the best of our knowledge, dialogue games have received few attention for practical applications in the human-computer interaction field. On a theoretical level, they have been seen as initiative-response units [8], and as structures capturing the different commitments created during dialogue [9,13]. On both practical and formal level, dialogue games have been conceived as recursive transition networks [14].

As [8,9], we propose to go towards a hybrid reactive/deliberative architecture where a theory of joint actions may serve as a 'semantics' to the interaction patterns described as dialogue games. Among the existing approaches, that of Maudet seems the most theoretically complete. It is the only approach that explicitly considers the entry and exit phases of a game as well as the multifunctionality of human dialogue [15] by differentiating two kinds of game (dialogue and communication games). This approach is the starting point for our model.

3 Implementation of the Methodology

In this section, we first describe the corpus. Next, we present the DIT^{++} annotation scheme [16] used to annotate our corpus. Then, we insist on the annotation process and its results. Finally, we say a few words about the extraction process.

3.1 Corpus

Our long-term goal is to build a *mixed-initiative assistant* [1] for information retrieval for the CISMeF system [17][1]. We use a formerly constituted corpus [18]. It is composed of dialogues of assistance on a medical information search task between a CISMeF expert and a user. Users are representative of the targeted audience of the future system since they were not medical specialists and they wanted to obtain answers about medical enquiries. Dialogues were recorded during the task where the expert and one user were facing a computer using the advanced search interface. This experimentation produced 18 dialogues between two experts and 18 volunteers. It contains circa 33 000 words and 1054 turns.

[1] CISMeF stands for "catalogue and index of French-language Health Internet resources" and is available at the URL `www.cismef.org`.

3.2 DIT^{++} Annotation Scheme

The use of language can be viewed as the performance of communicative actions since the speech act theory [10]. The DIT^{++} taxonomy takes a *context-change* (or *information-state update*) approach [19] to the interpretation of dialogue acts. Here, context can be viewed as the set of conditions which influence the interpretation or generation of utterances in dialogue [16]. This distinction is essential to take into account the fact that utterances in human dialogues often are *multifunctional* [15] and perform multiple communicative acts contrary to what traditional speech act theory says.

DIT^{++} is a *multidimensional* scheme based on a theoretically grounded notion of dimension [15,16]. Ten *dimensions* are distinguished. Among these, we could point out: the *task* (dialogue act which contributes to advancing the task or activity underlying the dialogue), *social obligations management* (dialogue acts that take care of social conventions such as greetings), *auto-feedback* (dialogue acts that provide information about the speaker's processing of the previous utterance), *time management* (dialogue acts signalling that the speaker needs a little time to formulate his contribution) and *own communication management* (dialogue acts to indicate that the speaker is editing his currently producing contribution). The taxonomy includes 88 *communicative functions* and consists of two parts: *general-purpose functions* itself consisting of three *hierarchies* (information seeking and providing functions, and action discussion functions) and *dimension-specific functions* (e.g., apology, turn grab). A *dialogue act* consists of a dimension and a communicative function. It is a context update operator construed by applying a *communicative function* to a *semantic content*. An utterance is segmented into *functional segments* which are annotated segments with zero or one communicative function per dimension.

The DIT^{++} framework proposes an application-independent taxonomy of functions for the analysis of human dialogue as well as the design of dialogue systems (and especially, the dialogue manager component). It has been shown that a multidimensional approach to dialogue annotation enables a more accurate analysis of human communication. Next, encouraging results were produced concerning the automatic recognition of DIT^{++} communicative functions by machine learning techniques. Eventually, DIT^{++} may be useful for both *interpretation* of verbal communicative behaviour, and *generation* of multifunctional utterance for ECA.

3.3 Annotation Process

Annotation was performed using the DIT^{++} taxonomy by means of the Gate annotation tool [20]. Four annotators worked on this annotation task. Each dialogue was annotated by two persons. One annotator performed the annotation for the whole corpus while the three others annotated one-third of the corpus.

The annotation process consists of two parts: (i) *Segmentation* of utterances into *functional segments* where a functional segment (FS) is "…a minimal stretch of communicative behaviour that has a communicative function. Such stretches do not need to be grammatically well-formed or contiguous, and may

Table 1. IAA for the labelling task for each dimension. R = Recall, P = Precision, F = F-measure

	Strict			Lenient			Average			
	R	P	F	R	P	F	R	P	F	**Proportion**
Allo-Feedback	0.63	0.54	0.58	0.66	0.56	0.61	0.64	0.55	0.59	1.19%
Auto-Feedback	0.77	0.8	0.79	0.8	0.83	0.81	0.79	0.81	0.8	9.44%
Contact Management	0.67	0.46	0.55	0.89	0.62	0.73	0.78	0.54	0.64	0.35%
Discourse Structuring	0.67	0.57	0.62	0.75	0.64	0.69	0.71	0.61	0.65	0.41%
Own Communication Management	0.43	0.49	0.46	0.47	0.54	0.5	0.45	0.52	0.48	5.31%
Partner Communication Management	0.86	0.91	0.89	0.86	0.91	0.89	0.86	0.91	0.89	1.09%
Social Obligations Management	0.43	0.69	0.53	0.48	0.76	0.59	0.46	0.72	0.56	1.18%
Task	0.84	0.85	0.84	0.86	0.87	0.87	0.85	0.86	0.86	68.30%
Time Management	0.75	0.8	0.77	0.81	0.86	0.83	0.78	0.83	0.8	9.93%
Turn Management	0.37	0.73	0.49	0.41	0.8	0.54	0.39	0.76	0.51	2.76%
Summary	0.77	0.81	0.79	0.81	0.84	0.82	**0.79**	**0.83**	**0.81**	

have more than one communicative function." [15]; (ii) *Labelling* of FS with zero or one communicative function per dimension.

The annotation strategy was "strictly indicator-based" [15]. Annotators were asked to mark communicative functions which are recognisable directly from features of the FS and given the context of the preceding dialogue.

3.4 Results of the Annotation Process

6343 communicative functions were produced during the annotation process with a total number of 5484 functional segments. We obtain similar results to [15] on the average number of communicative functions per FS for the selected segmentation strategy which is 1.16. A turn is in average composed of 2.60 FS thus confirming the multifunctionality hypothesis.

Since our annotation process is twofold (segmentation and labelling of functions), we performed an analysis of inter-annotator agreement (IAA) in terms of precision, recall and F-measure (F_1 score). These results come up in three categories: strict, lenient and average. These categories vary on how they consider overlapping annotations in the computation of precision and recall. The *strict* mode considers overlapping annotations as incorrect whereas the *lenient* mode considers them correct. *Average mode* takes the average of the two previous modes. The IAA for the labelling part does not take into account the taxonomic property of DIT^{++}: a `Check Question` and a `Yes/No Question` are considered as different as a `Thanking` and a `Yes/No Question` (which is obviously not true). Taxonomic metrics have been proposed for DIT^{++} but only take into account the labelling part and not the segmentation [16]. Hence, IAAs presented here are to be taken as the worst case, unless otherwise specified.

IAA on segmentation is strong since we obtain scores superior or equal to 0.93 in each mode. IAA per dimension for the labelling of communicative functions as well as the percentage of functions per dimension are presented in Table 1. First, we can realise that four dimensions stand out from the set regarding the proportion of functions: the *task* (68.30%), *time management* (9.93%), *auto-feedback* (9.44%) and *own communication management* (OCM, 5.31%) dimensions previously described. In broad outline, we could say that two-thirds of the functions are directed towards advancing the task motivating the dialogue, whereas one-third is directed towards management of the interaction process. IAA for these dimensions (except the OCM) exceeds 0.75 which we could qualify as a *reliable agreement*. The low score in the OCM dimension is due to a difference in the level of specificity of the communicative functions selected by the annotators. Actually, OCM functions form a branch in the taxonomic hierarchy. If we consider them equivalent, we reach a significant agreement of 0.67. IAA in other dimensions indicates a fair agreement that is penalised by a poor recall for the *social obligations management* and *turn management* dimensions. All in all, we obtain a global F-measure of 0.81 indicating a *reliable agreement* for the annotation task.

3.5 Interaction Pattern Extraction Process

Before addressing the extraction process, we established a *reference corpus* by randomly selecting one-third of the corpus (i.e., 6 dialogues out of 18). This reference corpus is kept as a validation basis. The 12 remaining dialogues were used for the extraction process. This process consisted in a *semi-automatic extraction* by one person of dialogue patterns. Patterns have been manually annotated in the corpus and *automatically extracted* by a tool that we design.

We focused on patterns on the *task* dimension for several reasons. This dimension is prevailing over other dimensions in terms of number of communicative functions. Next, its IAA is high (≥ 0.84). Last, functions in the predominant *time* and *own communication management* dimensions are *monologic* in the sense that they are concerned with hesitations and self-corrections from the speaker. They are not likely to be good candidates for the extraction of interaction patterns between two interlocutors.

We found 11 interaction patterns in the task dimension consisting of *general-purpose functions* and splitting into *3 categories*: information-seeking (6 patterns), information-providing (2 patterns) and action-discussion patterns (3 patterns). We mainly observed *initiative-response patterns* [8] consisting of adjacency pairs with preferred and dispreferred second pair parts. For instance, we observed patterns such as `inform/agreement` or `offer/declineOffer`. Each pattern takes the form of an *initiative dialogue act* followed by possible *response dialogue acts* with their observed proportion of occurrence. For instance, the suggestion pattern extracted starts with a `suggestion` act that can either be followed by an `acceptSuggestion` (preferred second pair part, 94.25% of the cases) or a `declineSuggestion` (dispreferred second pair part, 5.57% of the cases). Among the 11 patterns, ten are initiative-response units [8] and one is a three-step pattern, namely the correction pattern (e.g., `inform/correction/agreement`).

We manually extracted 439 occurrences of patterns in the task dimension in which 38.76% come in the information-seeking category, 27.56% in the information-providing category and 33.71% in the action-discussion category.

4 Modelling Interaction Patterns

In this section, we present our game framework based on the notion of *social commitments* as well as examples of games created from interaction patterns.

4.1 Game Framework

Social commitments are to be distinguished from the *private states* of agents such as belief and desire. In fact, social commitments are commitments that bind a speaker to a community [21]. These commitments are *public*. They are stored in a *commitment store* that is part of the public layer of the information state of a dialogue system. Social commitments are distinguished into *propositional* and *action* commitments. The former concerns commitments that do not deal with future action such as when A says "Paris is the capital of France" whereas the latter concerns commitments dealing with future action like "I will come at your place this evening.". We discern *dialogical* commitments from *extra-dialogical* commitments.

We express commitments as predicates with an arity of 4. An extra-dialogical propositional commitment takes the form C(x,y,p,s) meaning that x is committed towards y about proposition p. s refers to the state of the commitment that we explain below. The previous example produces the following commitments: C(A,y,$capital(france, paris)$,Crt) meaning that A is committed towards y to $capital(france, paris)$. For the sake of readability and since we are only considering dialogue involving 2 partners, we will ignore the second argument which specifies the partner. A *dialogical commitment* is contextualised in a game g and takes the form C_g(x,α,Crt) (meaning that x is committed towards y to do action α in the context of game g). Furthermore, it is possible to compose actions in commitments with the choice ($\alpha|\beta$), the conditional statement ($\alpha \Rightarrow \beta$) meaning that β will occur if α does and the persistent conditional action ($\alpha \overset{*}{\Rightarrow} \beta$) meaning that β will occur each time α does.

Finally, three operations are considered on commitments: *creation*, *satisfaction* and *cancellation*. This leads to the following five states that are possible for a commitment (inspired by [22]): (i) *inactive* (Ina) which is the state by default, (ii) *created* (Crt) which is the state right after the creation of the commitment, (iii) *cancelled* (Cnl) which is the state after a cancellation of a created commitment, (iv) *fulfilled* (Ful) which is the state after a satisfaction of a created commitment, and (v) *failed* (Fal) which is the state if a tentative to socially create the commitment has failed.

Our formalisation of games refines the one proposed by Maudet [9] by the addition of failure conditions, game-specific effects of dialogue acts and coherence constraints on the semantic contents of acts. Games represent *conventions* between interlocutors and are *shared bilateral* structures. They can be divided into two categories: *dialogue games* and *communication games*. Dialogue games are a particular kind of *joint activity* [10] *temporarily* activated during the dialogue for a *specific goal* (e.g., information-seeking game, action-seeking game, etc.).

Communication games are dedicated to more general interaction processes (e.g., mutual understanding, turn-taking, etc.) and are *permanently* activated.

A dialogue game is a 5-tuple $\langle \mathrm{En}_g, \mathrm{S}_g, \mathrm{F}_g, \mathrm{R}_g, \mathrm{Eff}_g \rangle$ where (i) En_g is a pair $\langle \mathrm{En}_g^I, \mathrm{En}_g^P \rangle$ defining the *entry conditions* of the *initiator* (En_g^I) and of the *partner* (En_g^P) which are conditions that must hold at the beginning of the game, expressed in terms of extra-dialogical commitments, (ii) S_g is a pair $\langle \mathrm{S}_g^I, \mathrm{S}_g^P \rangle$ defining the *success conditions* of the *initiator* (S_g^I) and of the *partner* (S_g^P) which are conditions that define a state of success in terms of extra-dialogical commitments, (iii) F_g is a pair $\langle \mathrm{F}_g^I, \mathrm{F}_g^P \rangle$ defining the *failure conditions* of the *initiator* (F_g^I) and of the *partner* (F_g^P) which are conditions that define a state of failure in terms of extra-dialogical commitments, (iv) R_g is a pair $\langle \mathrm{R}_g^I, \mathrm{R}_g^P \rangle$ defining the *rules* of the *initiator* (R_g^I) and of the *partner* (R_g^P) which are specifications of dialogue rules expressed in terms of dialogical commitments where constraints on the semantic contents of dialogue acts could be specified (as in [8]), and (v) Eff_g is a pair $\langle \mathrm{Eff}_g^I, \mathrm{Eff}_g^P \rangle$ defining the *contextualised effects of dialogue acts* in terms of the creation of *extra-dialogical* commitments for the *initiator* (Eff_g^I) and for the *partner* (Eff_g^P). S_g and F_g are conditions that motivate an exit of the game. As for communication games, their structures come down to *dialogue rules*.

As previously mentioned, game playing can be seen as a joint action [8]. Hence, game has an *entry*, a *body* and an *exit* [3,10]. Therefore, dialogue games need to be established. The communication game of *contextualisation* is dedicated to this task: we currently consider a simple version of this game consisting of two proposition phases (one for the entry and one for the exit) that must be explicitly accepted by the partner as in [13].

4.2 Examples of Games from Dialogue Patterns

We now present one dialogue game and one communication game created from the patterns extracted during the extraction process. For the sake of readability, we skip the create operation in conditional statements ($\alpha \Rightarrow \mathrm{create}(x, \mathrm{C_g}(y, \beta, \mathrm{Crt}))$ is equivalent to $\alpha \Rightarrow \mathrm{C_g}(y, \beta, \mathrm{Crt})$). Furthermore, dialogue acts take the form: $f\,(s, c)$ where f is the communicative function of the dialogue act, s the speaker that produces this act and c the semantic content. Hence we do not specify the dimension since we focus on the task one.

The choice question dialogue game is a representative example of all aspects of our framework and is presented on Table 2. We took the simplified semantic representation for choice questions from Larsson [23] but other semantics may be applicable (e.g., see [8]). A choice question q is viewed as an alternative question between propositions belonging to a set (e.g., "Would you like to add the keyword paludism, therapeutic or vaccine?") and takes the form: $\{?p_1, ?p_2, \ldots, ?p_n\}$. Three predicates relate to this kind of question. First, the **resolves** (p, q) predicate of arity 2 is true when the proposition p resolves q. Here, a resolving proposition is p_i with $1 \leq i \leq n$ (e.g., "Vaccine!"). Next, the **relevant** (p, q) predicate of arity 2 is true when the proposition p is about q.

Table 2. The Choice Question Game

Game g	Initiator (x)	Partner (y)
En_g		C(y,p,Ina) with **resolves** (p, q)
S_g	C(y,p,Crt) with **resolves** (p, q)	C(y,p,Crt) with **resolves** (p, q)
F_g	C(y,**fail** (q),Crt)	C(y,**fail** (q),Crt)
R_g	choiceQuestion(x, q)	choiceQuestion(x, q) ⇒ $\mathrm{C_g}$(y,answer(y, p)|answer(y, s)|execNegativeAF(y, q),Crt) with **resolves** (p, q), **relevant** (s, q) answer(y, s) ⇒ $\mathrm{C_g}$(y,answer(y, p)|answer(y, r)|execNegativeAF(y, q),Crt) with **resolves** (p, q), **relevant** (s, q), **relevant** (r, q)
Eff_g		answer(y, s) ⇒ C(y,s,Crt) execNegativeAF(y, q) ⇒ C(y,**fail** (q),Crt)

Table 3. The Agreement Communication Game

α	β
f(x,p)	C(y,agreement(y, p)|disagreement(y, p),Crt)
agreement(y, p)	C(y,p,Crt)
disagreement(y, p)	C(y,$\neg p$,Crt)

f ∈ {Inform, Answer, Agreement, Disagreement, Correction, Confirm, Disconfirm}

Here, a relevant but not resolving proposition would be $\neg p_i$ with $1 \leq i \leq n$ (e.g., "Not therapeutic."). Eventually, the **fail** (q) proposition indicates that an answer cannot be found by taking into account the current information state. The same simplifying assumption than [23] is done that the *resolves* and *relevant* relations are shared between interlocutors. Entry conditions specify that the partner must not already be committed on a proposition that resolves the question. Success conditions are reached when the partner is committed to a proposition that resolves the question. Failure conditions establish that a state of failure of the game happens when the partner is committed to the fact that he is not able to find an answer. Rules specify that the initiator is committed to play a `choiceQuestion` act. Once it is done, the partner is committed to play `answer` moves or an `execNegativeAF` move. This latter expresses the fact that a resolving answer can not be found. The rules state that the partner can give as many relevant answers as he can, and only one resolving answer. This enables to cover cases like "Not therapeutic. And yes vaccine !" where the first answer is relevant and the second is resolving. Finally, the effects precise that, *in the context of this game*, playing an `answer` move commits the speaker to its semantic content and playing an `execNegativeAF` move commits the speaker to **fail** (q).

Eventually, we present a communication game: the agreement game (Table 3). Rules take the form: $\alpha \overset{*}{\Rightarrow} \beta$. This game specifies that an addressee of an information-providing dialogue act can play an agreement or disagreement move after receiving this act. If he agrees, he is committed to the semantic content of the information-providing act, else he is committed to its negation. For example: "– We got 115 articles. (Inform) – Exactly (Agreement)"[2].

[2] All examples comes from our corpus and were translated from French to English.

Table 4. Results of the evaluation process (percentage of exchanges that fit a game)

Suggestion	Request	Offer	Agreement	Check Q	Posi-check	Nega-Check	Y/N Q	Choice Q	Set Q	Correction
84.85%	72.73%	78.13%	100.00%	68.92%	100.00%	66.67%	66.67%	87.50%	88.57%	100.00%

4.3 Discussion

An evident limitation of our work is that we only defined minimal dialogue games. The issue of how bigger interaction pattern emerges from these minimal games is left unaddressed for the moment. However, they may result of *compositions* of minimal joint action [8,9,10,13] such as *embedding*, *chaining* and *pre-sequencing* [10]. This problem is tightly connected to the idea of a *contextualisation game*. This is an area left for future work. However, we could point out that the minimal nature of the dialogue games that we defined associated with composition rules make the dialogue game formalism *flexible* and *reusable*.

The issue of the integration of dialogue games in the deliberative process of an agent is open. The model that we propose is a *normative* one. It can then be viewed as an *independent module* regulating the dialogical capabilities of an agent on both *interpretation* and *generation* of dialogue acts (thus simplifying the process of intention recognition). On the *interpretation* level, dialogue games make it possible to produce a set of conventionally expected dialogue acts whereas, on the generation level, they give a conventional motivation for the production of acts. The *independence* of the module is reinforced by the declarative nature of the formalism which is disconnected from the private states of the dialogue participants.

5 Validation

We conducted a study to validate the *structural property* of our model based on the reference corpus. Two voluntary persons of our laboratory were given the whole game definitions and were asked to annotate each *exchange* (i.e., a sequence of two (or more) functional segments produced by different speakers) with a game *if possible*. It was generated for each dialogue and for each game a ratio between exchanges that fit the game and those that do not.

Results of the validation process are presented in Table 4. The global conclusion is that 83.43% of the 350 exchanges of the reference corpus match a game that we defined. If we go into details, we see that scores are all beyond 66% indicating a reliable adequacy between dialogues being modelled and our games.

In addition, we investigated the 16.57% of mismatch cases and we identified 2 main categories: those related to the *inter-game structure* and those related to the *intra-game structure*. Mismatches related to the first category are twofold. On the one hand, the partner deliberately ignores the utterance of the initiator in approximately 10% of the cases. It is what Clark calls a *withdrawal from the joint project* [10] and is exemplified by: "– Would you like me to open this document ? (Offer) – Document 13 was interesting. (Inform)". On the other hand, the partner opens an *embedded game* that obsoletes the *parent game* in around

45% of the cases. It is illustrated by this example: "– Is there any documents that would suit you? – Well, is there the keyword 'prevention'? – No, we did not add it.". Mismatches related to the second category can also be subdivided into two parts. First, the predominant case (approx. 35%) is when the partner seems to take a shortcut in a bigger interaction patterns. It includes cases that has been called *indirect speech act* and are mostly appearing with Request game and CheckQuestion game (for instance: "– Can you formulate your information need or not? (CheckQuestion) – 'What is known about the evolution of current treatments of migraine' (Inform)"). Last cases appear when the partner alters its response from what is expected by the pattern to something that he is able and willing to comply with (approx. 10%), called an *alteration of the joint project* [10]. This can be illustrated by the following example: "– Then, you do not know the equivalent of Zomig? (CheckQuestion) – This is a product family.".

6 Conclusion and Discussion

We presented a data-driven methodology based on the study of human interactions to address the challenge of designing efficient human-agent interaction. This methodology is based on the constitution of a *matrix representation* of a corpus in several steps (collection, annotation, pattern extraction) that makes it possible to extract *interaction patterns*. The issue of the interest of dialogue patterns for dialogue management was raised. It turns out that minimal dialogue patterns can be viewed as the minimal unit of interaction, and therefore are very attractive to dialogue modelling. We then presented a framework to model such dialogue patterns based on theoretical work about *dialogue games*. This framework stands on the notion of *social commitments* that permits to specify dialogue games independently of the architecture of the agent. Hence, it makes it possible to envision the implementation of this model as a separate module with a normative role. We exemplified the specification of dialogue games from dialogue patterns by implementing all the steps of our methodology on a task-oriented corpus. We validate the games that we modelled against a reference corpus. The modelled games were able to describe appropriately the exchanges of the reference corpus.

Many interesting perspectives are possible. We focus our pattern study on one dimension. Future work involves the extraction and modelling of patterns on other dimensions. The model would include *multidimensional dialogue games* that could help producing multifunctional utterances. As stressed in this article, more work needs to be done on composition and contextualisation of dialogue games as well as implicit phenomena that appear in human dialogue.

References

1. Tecuci, G., Boicu, M., Cox, M.T.: Seven aspects of mixed-initiative reasoning: An introduction to this special issue on mixed-initiative assistants. AI Magazine 28(2), 11–18 (2007)
2. Ales, Z., Dubuisson Duplessis, G., Serban, O., Pauchet, A.: A methodology to design human-like embodied conversational agents. In: AAMAS 2012 Workshop, Human-Agent Interaction Design and Models (2012)

3. Mann, W.C.: Dialogue games: Conventions of human interaction. Argumentation 2(4), 511–532 (1988)
4. Grosz, B., Sidner, C.: Attention, intentions, and the structure of discourse. Computational Linguistics 12(3), 175–204 (1986)
5. Allen, J., Ferguson, G., Miller, B.W., Ringger, E.K., Sikorski-Zollo, T.: Dialogue systems: From theory to practice in TRAINS-96. In: Handbook of Natural Language Processing, pp. 347–376 (2000)
6. Rich, C., Sidner, C.: COLLAGEN: a collaboration manager for software interface agents. User Modeling and User-Adapted Interaction 8(3), 315–350 (1998)
7. Polanyi, L., Scha, R.: A syntactic approach to discourse semantics. In: Proceedings of the 10th International Conference on Computational Linguistics, pp. 413–419 (1984)
8. Hulstijn, J.: Dialogue models for inquiry and transaction. PhD thesis, University of Twente (2000)
9. Maudet, N.: Modéliser les conventions des interactions langagières: la contribution des jeux de dialogue. PhD thesis, Toulouse 3 (2001)
10. Clark, H.H.: Using language, vol. 4. Cambridge University Press (1996)
11. Cohen, P.: Survey of the state of the art in human language technology, pp. 204–210. Cambridge University Press, New York (1997)
12. Pulman, S.: The TRINDI project: Some preliminary themes. In: Proceedings of the Twente Workshop on Language Technology (1998)
13. Maudet, N., Chaib-draa, B.: Commitment-based and dialogue-game based protocols: new trends in agent communication languages. The Knowledge Engineering Review 17(2), 157–179 (2002)
14. Lewin, I.: A formal model of conversational game theory. In: 4th Workshop on the Semantics & Pragmantics of Dialogue (2000)
15. Bunt, H.: Multifunctionality in dialogue. Computer Speech and Language 25(2), 222–245 (2011)
16. Bunt, H.: The DIT++ taxonomy for functional dialogue markup. In: AAMAS 2009 Workshop, Towards a Standard Markup Language for Embodied Dialogue Acts, pp. 13–24 (2009)
17. Darmoni, S., Leroy, J., Baudic, F., Douyre, M., Piot, J., Thirion, B.: Cismef: a structured health resource guide. Methods Inf. Med. 39, 30–35 (2000)
18. Loisel, A., Dubuisson Duplessis, G., Chaignaud, N., Kotowicz, J.P.: A conversational agent for information retrieval based on a study of human dialogues. In: Filipe, J., Fred, A. (eds.) Proceedings of ICAART, vol. 1, pp. 312–317 (2012)
19. Larsson, S., Traum, D.R.: Information state and dialogue management in the TRINDI dialogue move engine toolkit. Natural Language Engineering 6(3&4), 323–340 (2000)
20. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M.A., Saggion, H., Petrak, J., Li, Y., Peters, W.: Text Processing with GATE (2011)
21. Singh, M.P.: Social and psychological commitments in multiagent systems. In: AAAI Fall Symposium on Knowledge and Action at Social and Organizational Levels, pp. 104–106 (1991)
22. Chaib-Draa, B., Labrie, M.A., Bergeron, M., Pasquier, P.: Diagal: An agent communication language based on dialogue games and sustained by social commitments. Autonomous Agents and Multi-Agent Systems 13(1), 61–95 (2006)
23. Larsson, S.: Issue-based dialogue management. PhD thesis, Department of Linguistics, Göteborg University (2002)

MAS-BPM: Multi-Agent System Bomber Problem Model

Zina Elguedria, Boutheina Jlifi, and Khaled Ghédira

L. SOIE, Stratégies d'Optimization des Informations et de la connaissancE
ISG, Institut Supérieur de Gestion, 2000 Le Bardo, Tunisie
zina.guedria@gmail.com, boutheina.jlifi@yahoo.fr,
khaled.ghedira@isg.rnu.tn

Abstract. Simulation has become unavoidable to help control societies with globalization and complexity. This term includes concepts as diverse as strategic decision support or staff training. Military interventions are characterized by using weapons systems increasingly sophisticated in highly diverse contexts. The use of force must control the risk of error to respond to military objectives. Multi-agent simulation is for a long time privileged for modeling and experimentation of complex systems. In this paper, we explore the challenge of simulating a system as complex as the Bomber problem with a MAS approach. Particularly, we demonstrate that MAS approach, models and simulates the Bomber Problem with efficiency. We illustrate our discussion with developed simulation results.

Keywords: Multi-agent System, Bomber Problem, Military domain, Simulation, Complex System, Interaction.

1 Introduction

1.1 Context and Motivation

An extensive literature is present in the military domain; however, the specificity of the bomber problem is that it has been specifically studied from mathematical component.

The problem of air defense emerged from the end of Worldwide War II as one of the most serious threats against the safety of a patrimony.

The first attempts to automate military defense systems were content to combine radar operations scattered with computers in order to achieve a defense system of the United States in Lincoln laboratory [1].

Since the simulation is a kind of representation of reality, then it is strongly related to the concept of model. Indeed, a model is an artificial reconstruction of a system, an entity, a phenomenon or a process. In addition, general military definition is to consider simulation as the real opposition: Anything that is not a real fight is a simulation. There are two main criteria to classify simulation techniques. Begin with the areas of application of simulation in the military domain which are as follows: Education and training; Prospect, studies and evaluations; Employment, planning,

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 61–72, 2013.

preparation and operation control, and decision support. The second classification criteria is composition or military level, as follows: defense system (strategic); the force system (operative); the fighting system (tactical); the weapon system (weapons system, information systems); subsystems of a weapon system [6]. The bomber problem can be classified as defense system and fighting system (strategic and tactical).

Multi-agent simulation is for a long time privileged for modeling and experimentation of complex systems as the bomber problem system.

1.2 Mathematical Formulation of the Bomber Problem

The Bomber problem BP [2-5] can be considered as a discrete time model in which a bomber must survive for t epochs before reaching the target where it will drop its bombs. At each epoch t with probability p, the bomber may encounter an enemy fighter plane.

The bomber has an integer number of Air-to-air anti-aircraft missiles in board, say n. We call these bombs in our approach, defense bombs. Indeed there are two objectives. The first is to accomplish the mission of the bomber successfully. The second objective is to optimize the number of defense bombs k in each encounter with an enemy fighter plane during an epoch t, say k (n, t).

The stochastic dynamic programming equation is clearly [5]:

$$V\ (0,x)=1 \tag{1}$$

$$V(n,t)=(l\text{-}p)\ V(n,t\text{-}l)+p\ (\mathrm{Max}(1\text{-}\alpha^{k})V(n\text{-}k,t\text{-}1) \tag{2}$$

Let k(n,t) be the optimal number of missiles to fire when the state is (n,t) and an enemy fighter is encountered. There are three obvious conjectures:

- (A) k(n,t) is nonincreasing in n;
- (B) k(n,t) is nondecreasing in n;
- (C) n-k(n,t) is nondecreasing in n.

Property (C) is easy to prove. Property (A) has been proved (although the proof is not easy. See [2, 3]. Property (B) has never been proved. But it has been tested extensively by numerical solution of the dynamic programming equation. This resolution of (B) remains an intriguing open problem that has been outstanding for 50 years.

The bomber problem is subject to a set of constraints related to the number of defense bombs as follows:

First, the encounter with enemy fighter plane constrains the use of defense bombs.

p is the probability that a meeting is true. Hence, we need to minimize the probability for each period t, thus we must minimize p (n, t).

If p is equal to 0 then there is no meeting and the number of defense bombs remains unchanged.

If the bomber shoots k of its n bomb, he has a survival probability p = $(1\text{-}a^{k})$. Therefore we need to maximize p while minimizing k. On the other hand, we must maximize the probability that the aircraft encountered destroy all his enemies and accomplish its mission. Thus, we have to minimize p (Max $(1\text{-}\alpha^{k})$).

If the number of defense bombs is exhausted there will be two assumptions: Hypothesis 1: the number of bombs is static. The bomber tries to accomplish his mission by avoiding encounters with the enemy fighter plane.

Hypothesis 2: the number of bombs is renewable. The bomber must seek the nearest and the most secure source.

1.3 Bomber Problem as a Complex System

The Bomber problem is unsolved despite his appearance date since the 1960s [4]. It is classified in the heading of research problems unsolved by Richard Weber [4]. We propose to tackle it by bringing together different axes and research areas such as multi-agent systems, complex systems, and military simulation.

The Bomber problem can be considered as a complex system. A complex system can be defined as a system composed of subsystems, including design and operation involving different trades, and no one can apprehend as a whole. The design of a complex system therefore requires methods and tools to ensure compliance components of subsystems and system specifications eventually throughout their realization: quality of service and capacity of new subsystems. Such is the case of our complex system that involves information technology through multi-agent system simulation in a military application.

2 A Multi-Agent System for the Bomber Problem

2.1 Basic Concepts

Having fixed from the beginning that our goal is to design a distributed system to solve the Bomber Problem. Our distributed model of resolution and simulation has essentially the strength of multi-agent systems, which allow the resolution of problems in terms of cooperation, competition and negotiation in a society of agents. The basic idea of our MAS model is to design: a Base agent who will be the coordinator throughout the resolution process; a set of Craft agents who will be the bombers that are designed to perform one or more tasks assigned to them by the Base agent.

During the accomplishment of its mission, the Craft agent may encounter Enemy agents. This confrontation involves triggering a battle between them. The Craft agent can also carry out its tasks as a team, while forming coalitions. In the following, we detail our model.

2.2 Agent Structure

Each agent has a structure: acquaintances (the agents that it knows and it can communicate with), a local knowledge composed of its static and dynamic knowledge and a mailbox where it stores the received messages that will be later processed one by one.

Base Agent

Acquaintances. This agent communicates with all Craft agents.

Knowledge. Knowledge of a Base agent can be classified into two categories:
Static knowledge such as:

- T-Miss :the type of mission that can be either individual or collective,
- mDist : the matrix of distances between the different Craft agents and different targets,
- lMiss: the list of missions assigned to a Craft agent i.

Dynamic knowledge such as:

- (xi, yi): the position of a Craft agent,
- Nbrdead: the number of dead Craft agent during a mission,
- NbrDestroyed: the number of goals successfully destroyed by Craft agents,
- NbrMove: the number of move made by a Craft agent,
- baseRad: the basis of observations collected from different Craft agents the base agent.

Mailbox. The Base agent has a mailbox where it stores different messages it exchanged with other agents.

Craft Agent. This agent is a bomber. There are N instances of this type of agents in our system. It is a BDI[1] agent. It is defined by its:

Acquaintances. It communicates with all Craft agents and the Base agent.

Knowledge. Knowledge of Craft agent can be classified into two categories:
Static knowledge such as:

- its identificatior i,
- xi and yi : the coordinates of the position i in a map,
- Dmin (i, j): the minimum distance between two Craft agents to get into the same a team,
- QBdi : the capacity of defense bombs gauge of a Craft agent i,
- QBvi : the capacity of bombs gauge to destroy targets by a Craft agent i,
- QFi: the capacity of fuel gauge of a Craft agent i,
- tMiss : the type of mission that can be either individual or collective,
- lMiss : the list of missions assigned to a Craft agent i,
- lDepi : the list of dependent Craft agents to a Craft agent i,
- lSupi : list of supervisor Craft agents of a Craft agent i,
- mDist: the matrix of distances between the different Craft agents and different targets.

Dynamic knowledge such as:

- (xi, yi) : the position of a Craft agent,
- lEqui: the list of the equivalent Craft agents of the Craft agent i.
- lTeami: list coalitions or teams to which the Craft agent i can belong as a supervisor, and from whom it will propose coalitions with the other Craft agents.

---

[1] Belief, Desire, Intention.

- lHistMi: the history list of the missions carried out by the Craft agent i with mention of the result(success, failure).
- lHistFi: the history list of fights made by Craft agent i with the mention of outcome of the fight(victory, defeat, escape).
- jlifei: the level of gauge of life of a Craft agent i.
- jBvi: the level of bombs gauge to destroy targets by a Craft agent i.
- jBdi: the level of defense bombs gauge of a Craft agent i.
- jCi: the level of fuel gauge of a Craft agent i.
- aRadi: the global view of the Craft agent i, the observations that it transmits to the Base agent.
- lEtati: the list of the states of Craft agents met or with which a Craft agent i communicated. The state of a two Craft agents can be equal to "waiting" if they can or not belong to the same coalition (or equips), "enemy" if they are two enemies or "friend" if they belong to the same team.

Mailbox. The Craft agent has a mailbox where it stores different messages it exchanged with other agents.

We notice that Craft agent's behaviors and Base agent behaviors will later process one by one in global dynamic section.

Enemy Agent. An enemy agent is similar to a Craft agent with a difference that it is reactive and does not belong to the communication system of Craft agents and Base agent. The only interaction between this agent and the others is in a fight. An enemy agent can be described as follows:

- It is a reactive agent.
- Its knowledge is static.
- The enemy agent is destroyed due to a number of successful shots made by a Craft agent during a fight.

2.3 Global Dynamics

In this section, we will describe the global dynamics of the multi-agents system set up: First we start with Base agent behavior. Then we tackle the Craft agent behavior (a mission). In case something happens and requires modification of the Craft agent behavior, it must choose one of its exception scenarios. Principal scenarios of the process and We introduce also, the most important messages exchanged between the various agents to then incorporate them in their behaviors.

Base Agent Behavior. After its creation, the Base agent initiates the process. It is responsible of the creation and initialization (static and dynamic knowledge, acquaintances, and distance matrix) of a number of Craft agents sufficient to accomplish the available missions. It affects missions to various Craft agents. It collects observations from Craft agents and stores them in the radar database denoted by baseRad. The baseRad will be sent to the Craft agents involved.

Craft Agent Behavior. As in real world mission, the Craft agent behavior is a mission. The mission typically includes a sequence of events that must be completed to reach a successful conclusion of the mission. In the beginning, the Craft agent receives its list of missions. It seeks then, the optimal route to reach its target area. Once arrived, the Craft agent bombards the target. If it no longer has a mission in its list and if nothing happens during these phases, the mission is successfully completed

In the case where the list of missions contains more than just a single mission, the Craft agent had to complete all figuring mission. Once mission is completed, the Craft returns to the military base to renew its gauges and get new missions.

In real world of bomber, the aircraft may encounter enemy fighters, it may also have to ask for help from other bombers and accomplish mission in team. Whatever the case, if something happens and requires modification of planed mission; the Craft agent must choose one of its exception scenarios.

Scenario 1: Encounter with an Enemy and Fight. Encounter with an Enemy agent during accomplishment of the mission of the Craft agent may engender a fight. Both Enemy agent and Craft agent are characterized when fighting by:

- Life gauge jlifei: This gauge represents a number of successful shots needed to touch the adversary agent. The exhaustion of this gauge involves the complete destruction of the concerned agent.
- The defense bombs gauge jBdi: This gauge is the number of defense bombs, which an agent may use during Fight.

The life gauge and defense bombs gauge of an Enemy agent are determined randomly when meeting with a Craft agent. In fact, the Enemy agent is part of the external environment of our model, which implies that its behavior is unpredictable, as well as its knowledge and performance.

The Fight scenario between Enemy agent and Craft agent occurs in the following steps: First, the Craft agent checks its gauges level; if the gauges level are insufficient to accomplish a fight (jlifeior jBdi $\leq$ 1), the Craft agent sends a <ReinforcementMsg>to other Craft agents and the Base agent to ask for reinforcement, else with sufficient gauge level, the fight begin. Second, Craft agent pulls a defense bomb on its enemy fighter; The Craft agent defense bomb gauge is decremented by 1. If the shot successfully touches Enemy agent its jlifei is decremented by 1; else nothing happens. The previews step is repeated in turn between the Craft agent and the Enemy agent until one of them is destroyed. Finally, once the life gauge of an agent is zero, it is destroyed. The fight can be concluded with escape, victory or destruction of one of agent fighting.

Scenario 2: Reinforcement. A Craft agent having low gauges (jViei =1 oujBdi, jBvi, jBci $\leq$ 1), sends a<ReinforcementMsg> to the Base agent and other Craft agents for two situations:

- During a bombing mission;
- During a fight.

When receiving <ReinforcementMsg>from a Craft agent, the Base agent assigns the class “dependent” to it. It then sends a message to other Craft agents <askHelp>in the neighborhood of the Craft agent who needs reinforcement (distance ≤ dMin). The Craft Agents receiving a message <askHelp>have the possibility to reply with two sorts of message: It can refuse for three reasons, namely being in a bombing mission, a fight or the Craft agents already belongs to another team. It can also accept. Then, the Craft agent is available and confirms the possibility of forming a team. It enters a process of negotiation with other Craft agents to schedule the team members. After forming the team, Craft agents in “supervisors” class complete the fight interrupted and then the mission, while independent agents seek the path containing the least of danger to return to the base.

Scenario 3: Mission in Team. Craft agent responding with accepts to an <askHelp> message enters in a process of negotiation with other Craft agents to schedule the team members. The Craft agent who sent the request for reinforcement is marked as a dependent agent. The nondependent agents transmit to the other agents in their neighborhood, a message containing the levels of all their gauges. As answer, aCraft agent receives two types of messages:

- If state i is less than the state j, the answer is: I am dependent <UpdDep>.
- If state i is better than the state j, the answer is: I am supervisor <UpdSup>.

In case of receiving a <UpdDep>and when it is already a supervisor of other Craft agents, the Craft agent sends a message to his dependents <UpdSup>. They update the hierarchy of the team in their knowledge bases. The negotiation process ends when after an exchange of messages, the hierarchy does not change and only one Craft agent or set of equivalent supervisor Craft agents lead the team.
Supervisor Craft agent decides who will complete the mission or the fight. As for the final dependent Craft agents (in the bottom of the hierarchy) are trying go back to the military base with the least contact with the environment.

Scenario 4: Radar. The radar scenario is based on the mechanism of MANET[2]. Each Craft agent has radar. Once it detects an Enemy, it sends a message to the Base agent <aRad>which collects the observations of all Craft agents. Subsequently, the Base agent sends a message <BaseRad> to Craft agents located in neighborhoods of these detected enemies.

3 Experimental Design

The objective of this section is to demonstrate the efficacy of the deployment of our multi-agent approach in the context of BP simulation. In fact, our goal is to see through simulating scenarios that have been described, that the missions have been fulfilled with good satisfaction as soon as possible and that coordination led to good

[2] Mobile Ad Hoc Network.

results. To achieve this goal, in the same way that it is necessary to have a methodology when it is a question of conceiving complex applications, it is necessary to have tools and generic components making it possible to consider the reuse and facilitate the development of the applications containing multi-agents systems. To carry out a simulation MAS, several platforms exist. The implementation of our approach was realized using the multi-agent platform JADE based on JAVA language. In our implementation, the type of environment used is a grid whose construction is intuitive.

In our implementation, we note that a simulation run is a mission and each agent can move randomly and independently on a grid (from box to box), which is large enough to simulate a battlefield. Finally, we describe the results obtained from a set of simulations performed according to different configurations.

3.1 Communication and Interaction

Communications and interactions are the focus of our proposed model. We can identify two types of interactions: those that occur between Craft agents and which take part directly in the evolution of the model and those that occur between Craft agents and Base agent.

- Base-Craft interaction: interaction is initiated if a Craft agent wants to perform an action. This interaction is successful if the Craft agent performs desired action. If action type is a move, after a successful interaction, the agent who was in a position it teleports to a random position. In our simulation, a Craft agent can perform 12 actions. Include: moving, bombing, and shooting an enemy.
- Craft-Craft Interaction: An interaction is initiated if a Craft agent wants coordinate with another Craft agent. This interaction is successful if a Craft agent accepts a request for coordination and invites it to a meeting position to start the mission. In our implementation, a Craft agent may initiate team formation by sending a request to another Craft agent to accomplish a mission. Craft agent can: accept, confirm, refuse, or terminate coordination.

3.2 Implementation

A simulation configuration is: Craft agent number/Enemy agent Number/Target Number. The results of the implementation of MAS-BPM are shown in the table 1 according to various configurations.

Table 1 shows the average amount of time taken to complete six simulations. Time shown is in minutes. It also shows the number of coordination within six simulations. Then it shows the number of interaction within six simulations. The six tested simulations were with respectively 5, 10, 20, 25, 50 and 100 Craft agents. These configurations were in two types of grid; indeed we used the first grid with a size of 10x10 and the second one with size of 20x20.

Table 1. Simulation Results with grid size variation

Configuration	Duration		Coordination		Interaction	
	10x10	20x20	10x10	20x20	10x10	20x20
5/5/5	2	10	76	363	736	2872
10/8/8	2	14	135	178	944	5206
20/8/8	3	13	114	257	974	4678
25/10/10	3	17	138	321	1336	4266
50/80/50	17	43	244	554	1271	10568
100/80/80	21	86	396	1763	1724	8559

Time Effect

Simulation Results. Figure 1 plots the average amount of time taken to complete six simulations. The two curve graphs illustrate simulation duration in two types of grid. First curve graph starts in two minutes and ends in twenty one minutes (10x10 grid) beside, second curve graph starts in ten minutes and ends in eighty six minutes(20x20 grid).

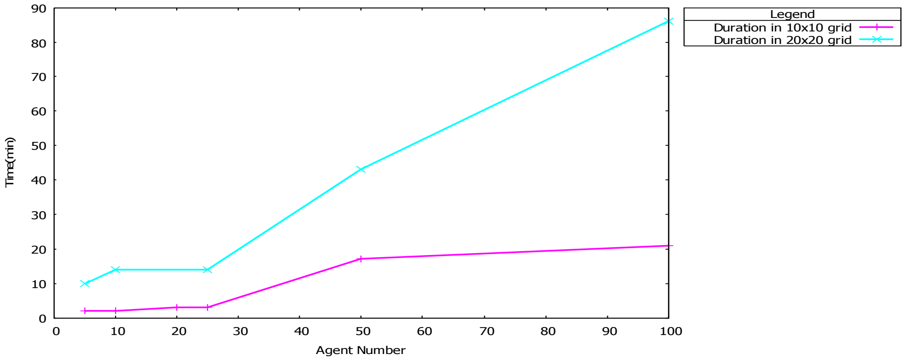


Fig. 1. Simulations duration with grid size variation

Discussion. Varying the grid size influences the variation of the CPU time during simulations. As proof, the first simulations with a grid of size 10x10, last a maximum of 21 minutes while simulations with a grid of size 20x20 are expected to exceed 20 minutes starting from 30 agents to achieve 86 minutes with 100 agents. It should be noticed that the simulation within a grid of size 60x60 lasts 300 minutes.

Coordination and Interaction Effect

Simulation Results. Figure 2 plots the number of coordination carried out during six simulations. The two curve graphs illustrate coordination number in two types of grid. Both curve graphs start zero coordination the first ends in 396 coordination (10x10 grid) beside, second curve graph ends in 1763 coordination (20x20 grid).

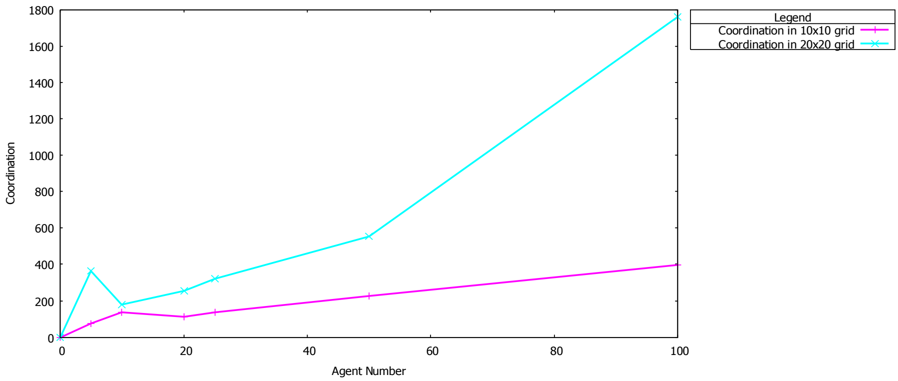


Fig. 2. Coordination number with grid size variation

Discussion. The simulation results demonstrate, as show the curves, in a grid of size 10x10, the effect of coordination is more visible when using a small number of Craft agent. In fact, coordination ensures the fulfillment of the mission and reduces the number of destroyed agents during the mission. Starting from 50 Craft agents, we notice an explosion of coordination number in all simulations tested. In this case, we opt the individual mission accomplishment. In against part, in a 20x20 grid size, the effect of coordination is more visible starting from 50 Craft agent or higher.

For a briefness issue; we have omitted the curve of the number of interaction. The simulation results demonstrate that in a grid of size 20x20, communication and interaction are more important than in a grid of size 10x10. In fact, communication and interaction ensure Craft agent position, enemy position and target position exchange between Craft agents. Indeed, in a small environment, there is no need for an important communication and as shows the figure 1 mission are briefly completed.

Success Ratio

Simulation Results. Figure 3 plots the number of destroyed Craft agent during six simulations in two types of grid. Figure 4 plots success ratio of eleven simulations according to two type of configuration the first one is 25/10/10 and the second one is 50/80/50.

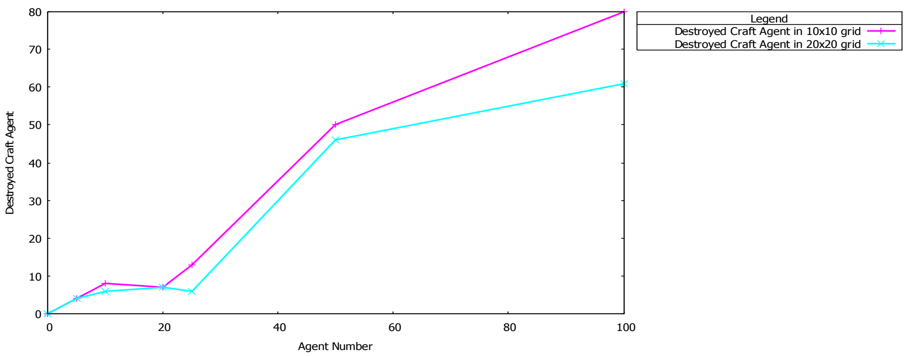


Fig. 3. Destroyed Craft agent with grid size variation

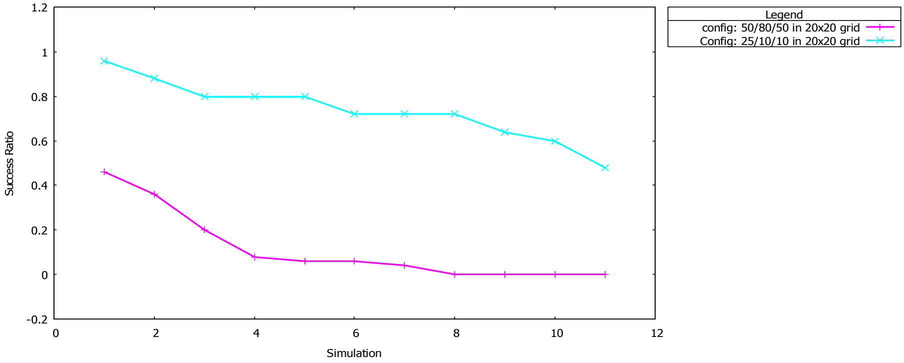


Fig. 4. Success Ratio

Discussion. The simulation results demonstrate, as show the curves in figure 3, in a grid of size 10x10, destroyed Craft agent is more important than in a grid of size20x20. In fact, the important communication and interaction shown in previous section in a 20x20 grid ensure more efficiency when carrying out a mission. Indeed, in a small environment, encounter with enemy is more randomly since there is no important communication between Craft agent to prevent each other when encountering an enemy.

We notice for all configurations, as show the curves in figure 4 the success ratio of a mission depends essentially on the choice of the number of Craft agents as well as good prediction of the number of Enemy agents that will encounter a Craft agent during its mission. As proof, the first simulation with 25 Craft agents against 10 Enemy agents gives success rates ranging from 48% to reach 96%. Beside, when choosing 50 Craft agents against 80 Enemy agents we had a less important success rate ranging from 4% to 46% and sometimes we notice a failure as a mission result, which justifies the importance of good preparation for the fight environment.

4 Conclusion and Perspectives

We have developed a society of agents, dynamically created and cooperating not only to provide a simulation of the Bomber problem but also to accomplish bombers missions and minimize the loss in terms of destroyed aircraft. An implementation and experiments were performed using JAVA language in the context of multi-agent systems. The experimental results show that our simulation deploys with efficacy the BP. In fact, in our case the MAS is replacing a method of optimization. That's why we obtained good results, namely, fewer destroyed Craft agent with more bombarded goals in a reduced amount of CPU time. Craft agents exploit their environment through communication; perform their mission based upon coordination and cooperation through a process of negotiation. Our simulation allowed us to have a good idea about the most effective configuration. It should be emphasized that the originality of our approach consists not only in the fact that the Bomber problem is not resolved but also in the fact that it brings together different research areas such as multi-agent systems, complex systems and simulation in a military application.

As perspectives, we first propose the necessity to add an optimization method in Craft agents, when searching the target and optimizing the number of defense bombs to fire. Then, we implement our approach using JADEX or Jade4BDI, which are two platforms for the implementation of BDI agents. Finally, we will take into consideration all constraints such as time constraint.

Acknowledgement. We acknowledge and extend our heartfelt gratitude to Pr. Edward Tsang who introduced the Bomber Problem to us and gives a special care to help us to the accomplishment of this work.

References

1. Lemnios, W.Z., Grometstein, A.A.: Overview of the Lincoln Laboratory Ballistic Missile Defense Program. Lincoln Laboratory Journal 13(1), 9–32 (2002)
2. Klinger, A., Brown, T.A.: Allocating unreliable units to random demands. In: Karreman, H. (ed.) Stochastic Optimization and Control, pp. 173–209. Wiley, New York (1968)
3. Samuel, E.: On some problems in operations research. Journal of Applied Probability 7, 157–164 (1970)
4. Weber, R.R.: A problem of ammunition rationing. In: Radermacher, F.J., Ritter, G., Ross, S.M. (eds.) Conference Report: Stochastic Dynamic Optimization and Applications in Scheduling and Relatedfields, p. 148. Facultät für Mathematik und Informatik, University of Passau (1985)
5. Weber, R.R.: ABCs of the bomber problem and its relatives. Annals of Operations Research (2011), doi: 10.1007/s 10479-011-10152
6. ICHEAr 2009 « l'institue des hautes études de défense nationale de la France: Avenir de la simulation pour l.entraînement des forces: quels bénéfices pour le fonctionnement et quelles limites? »45e session nationale (2009), http://www.ihedn.fr/

Agent Perception Modeling for Movement in Crowds

Katayoun Farrahi[1], Kashif Zia[1], Alexei Sharpanskykh[2], Alois Ferscha[1], and Lev Muchnik[3,4]

[1] Pervasive Computing Institute, JKU Linz, Austra
[2] Department of Computer Science, VU University Amsterdam, Netherlands
[3] School of Business Administration, The Hebrew University of Jerusalem, Israel
[4] Beacon Tech Ltd.

Abstract. This paper explores the integration of a perception map to an agent based model simulated on a realistic physical space. Each agent's perception map stores density information about the physical space which is used for routing. The scenario considered is the evacuation of a space given a crowd. Through agent interactions, both in physical proximity and through distant communications, agents update their perception maps and continuously work to overcome their incomplete perception of the world. Overall, this work aims at investigating the dynamics of agent information diffusion for emergency scenarios and combines three general elements: (1) an agent-based simulation of crowd dynamics in an emergency scenario over a real physical space, (2) a sophisticated decision making process driven by the agent's subjective view of the world and effected by trust, belief and confidence, and (3) agent's activity aimed at building relationships with specific peers that is based on mutual benefit from sharing information.

1 Introduction

Increasing abundance of mobile communication and sensor technologies accompanied by the evolution of mobile computational power suggests that these technologies may alter the very nature of human communication schemes and information diffusion dynamics. This may have a profound effect on the patterns of human behaviour, especially in the situations in which people rely on the availability and quality of information.

In everyday situations the existing infrastructure provides information of adequate quality and in a timely manner and people are typically capable of accessing the information and adjusting their activities accordingly. However, many of such pre-deployed systems are not as useful and effective in emergency situations, like natural disasters or terror attacks. The usual information delivery channels may be disrupted, and of importance – the information consumers may require is at an entirely different rate of update and level of detail when coping with their specific situation. A balance between providing a broad image simultaneously with the information required locally is crucial. In particular, a successful evacuation may

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 73–84, 2013.

depend on the underlying physical infrastructure as well as on coordination. The physical infrastructure adjustments tend to be quite costly. Agent based models provide a reasonable solution for agent behaviour prediction and analysis.

Agent Based Modeling is an important tool, particularly relating to recent developments in Computational Social Science [1]. For effective, realistic agent based modeling frameworks, it is of great importance to integrate cognitive models for social simulation [2]. With the recent revolution in ambient intelligence and the increasing trend of social media usage for interaction, the need of exploring social networks in social simulation is evident. Towards this, a social simulation should model the “process” influencing the buildup of a social network. Consequently, it should also analyze the “structure” a networks evolves into based on the environment and parameters describing the process. As indicated by Alam and Geller in [3], structure of a social network emerges based on the modalities of the process. One of the most important modalities of a process in social simulation (and social networking within it) is the “connectivity” between the interacting agents. The connectivity of the agents is dependent on communication as well as spatial features.

In this paper, we model an agent based framework on a real physical space. We are simulating the evacuation of the space given a crowd of agents. Agents make routing decisions towards points of attraction, where we assume once these points have been reached the agents are safe. The routing decisions are based on a novel cognitive decision model, which integrates belief, trust, and confidence and is based on agent communications. Communications occur based on the physical proximity of other agents and distant communications, simulating phone conversations. The entire process is encapsulated by the agent perception map, which is the main contribution of this work. Overall, the agent perception map (which is the agent’s perception of the density of the world, or physical space) contains the routing information which is updated based on agent communications.

The contributions of this paper are: (1) a novel cognitive decision model based on trust, belief, and confidence, (2) a realistic simulation framework for crowd evacuation in an emergency scenario, (3) the encapsulation of a cognitive decision model for agent routining based on the forumulation of an agent perception map.

Based on our simulation results, we investigate the validity of our model. Some examples of our findings are that (1) the full communication model results in a higher number of pair-wise agent trusts given higher degrees of trust, (2) agents which communicate locally only are able have higher accuracy in their density perception of the world, though agents with a full communication mechanism are able to perceive more density information about the world (have more perception map information) though with slightly less accuracy, and (3) the node degree distribution of the evolved trust network exhibits the same overall shape as a real mobile phone communication network.

2 Related Work

From a networking science perspective, a number of related studies follow Kleinbergs generative model [4] that explored the emergence of spatially embedded networks and their searchability. In particular, Liben-Nowell et al. [5] investigate the functional dependence of the probability of tie existence on the distance between LiveJournal users. The effect of a distance on the cellular communication patterns was explored by Lambiotte et al. [6] at a customer level and by Kings et al. [7] at an inter-city level. Adamic and Adar [8] explore the geographic properties of e-mail exchange networks within a company, while Mok and Wellman [9] focus on how the frequency of offline face-to-face interactions decays with distance. However, these studies did not directly address the specifics of the information benefits, geography, details of cognitive processes or the evolution of trust relationships between the peers typically focusing on the network structure and the distance as the fundamental underlying mechanisms of the suggested generative models.

There are many related works in the agent-based modeling community. In many existing models (e.g., [10]) crowd dynamics are considered from a lattice gas perspective by representing the systems actors by particles interacting through forces and fields. Although such models are highly scalable, they ignore (complex) internal dynamics underlying the decision making of actors, and, thus, cannot be used in cases for which rich cognitive and affective representations are required (e.g., reasoning, human decision making).

In addition to the importance of integrating the cognitive models into social simulation in general [2], the importance of human behavioral modeling (cognitive and social) specific to the emergency situation has already been noted [11]. However, in many of these efforts, the cognitive decision making rules are either very simple [12], or investigated only on an operational level [13]. The strength of our model is mapping cognition based reasoning on the decision making related to an evacuation situation from a city. We present explicit relationships (based on well-established neurological and psychological theories) between intentions and emotions in decision making.

A few studies [14,15] investigate the effects of information spread and emotions in crowds. In these studies, no ambient devices for communication over distance are used. Furthermore, in contrast to the model proposed in this paper, these studies do not consider trust relations and evolution of social networks.

In [16] an agent-based decision-making model in the context of crowd evacuation is proposed, which integrates existing neurological and cognitive theories of affective decision making. In contrast to our model, this model does not use crowd density as a decision criterion, and does not consider the evolution of the social networks. Furthermore, simulation in this study was performed on a smaller scale.

3 Agent Based Model

Our agent based model is simulated using Repast for High Performance Computing (Repast HPC) [17] for high performance distributed computing. It consists of multiple models defining space, mobility, perception, communication, and decision making, all formalized in the following sections.

3.1 Physical Space

Cells. The physical space in which agent movement takes place is taken from a neighbourhood in a real city in Linz, Austria. A raster image of the map is incorporated into the model by first reducing it to an area of 500 cells, where each *cell* is a unit of space equivalent to 1.25×1.25 m^2 in reality. The space referred to as a cell is later used for modeling individual agent mobility and for assuring two agents do not 'step on one another' or overlap in space.

Sectors. The map is segmented into 25 equally sized *sectors* for processing, where each sector is simulated by a single processor (25 processors in total). The area simulated and the division into sectors can be seen in Figure 1.

Map Generation. For agent mobility on the physical space, the map has to be converted to a binary grid. In order to achieve this, streets are selected as walkable areas (agents can move here), and all other areas are considered to be non-walkable (agents cannot move here). A smoothing algorithm was run over the space, first horizontally, then vertically, to counter inconsistencies in raster. The smoothing function is a low-pass filter, with filter coefficients equal to the reciprocal of the span.

Points of Attraction. For the emergency scenario simulation, we consider "points of attraction" (PoA), where an agent is considered to be "safe" once having reached these points. These PoA's are used in order to evaluate the decision making and mobility modeling. The PoA's can be seen in Figure 1 (b), illustrated by the red boxes in the corners of the space modeled. In this paper, the coordinates of the PoAs are provided manually, but in future work, our approach could easily be extended to handle random PoA generation.

3.2 Mobility

Our agent based model has the capability of modeling different transportation modes though this would result in an extra dimension of complexity in the results, so we chose to address one mode of transport for this work. Every agent requires individual basic routing information in order to move independently, assuming that all agents are pedestrian. In order to achieve this, we use the cell floor field method [18] to transfer information to the agent occupying a space at a given instant in time. This is the main reason for defining a cell (in Section 3.1). Each cell contains three variables, which are used for agent movement decision making. These are as follows.

1. **Direction**: The directions of each of the PoAs from the current cell. We refer to this feature as the direction of motion (DOM). Each DOM ranges from 0 to 355.99, calculated as the relative angle between the current cell and the cell containing the PoAs.
2. **Distance**: The physical distance to each of the sectors containing PoAs from the current sector, referred to as the hop count (HOPC). The HOPC is computed as the number of cells between the current cell and the cell of the PoAs.

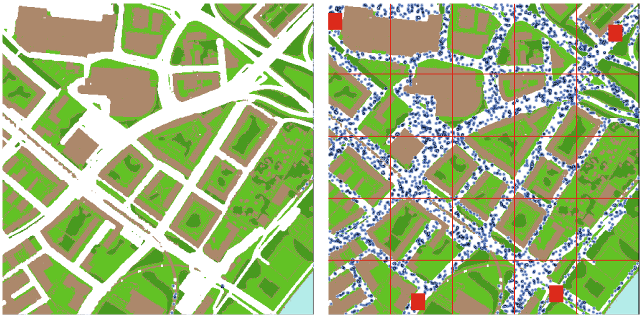


Fig. 1. View of the physical space (also referred to as the world) divided into 25 sectors. The sectors are necessary for efficient processing. Each sector is processed by an individual processor, and the agent decision making and perception of the world (described in the next section) is based on these 25 sectors. All white patches represent streets and are walkable by agents. All other are non-walkables by agents. In (b), a view with 5000 agents distributed uniformly over the space is visualized, with the points of attraction shown near the corners with red boxes.

3. **Route**: The route is the sequence of sectors that need to be traversed by agents in order to reach each of the PoAs.

An agent makes a routing decision based on a PoA selection, formalized next.

Routing Decision. If we define a point of attraction, poA, as a series of sectors forming a route (R), we can formalize this as $R = \{ID_{j_1}, ID_{j_2}, ...ID_{j_N}\}$ where ID_{j_i} is the identifier for sector j_i. The subscript of j denotes the index of the process in the route j. We assume N processes form a given route. We compute the average density for each route as

$$\rho(poA_j) = \sum_{e=1}^{N} \rho(ID_{j_e})/N. \tag{1}$$

The average density is then also weighed in conjunction with distance (d) as:

$$\omega(poA_j) = \rho(poA_j) * d(poA_j) \tag{2}$$

The point of attraction selected, poA^*, is chosen to be the one with the minimum weight over the route. Formally,

$$poA^* = poA_j with min(\omega(poA_j)). \tag{3}$$

Speed. We assume an agents' speed is affected by the density in its current vicinity. Therefore, an agent's speed is density based where the agent is assumed to know the density of it's current region (or sector). The formulation for speed is based on the free flow speed and is given by the following equations:

$$speed\_on\_density = v_o * (1 - N_{agents}/N_{walkables}) \tag{4}$$

$$speed = max\{v_{min}, speed\_on\_density\} \tag{5}$$

where $v_o = 1.36$, $N_{agents}/N_{walkables}$ is the density of a sector and $v_{min} = 0.0136$. Note that speed is not constant and is defined by *speed_on_density*.

3.3 Perception

Each agent maintains a perception of the surrounding world and updates it by collecting information through his "sensors" (i.e. personal observation) and receiving information via communication from his peers. The perception may be correct or not. The peers may transfer information by physical proximity interactions, or based on distant communication means (for example phone calls) with trusted peers (or friends).

Given this overview, we define a perception map for each agent, where each agent has a perception of the density in each of the sectors in the world, where the sectors are the 25 shown in Figure 1. Therefore each agent has a perception map containing 25 density values, which are updated continuously over time. In the case of an emergency scenario, the critical feature is the density so that an agent can reach an exit as quickly as possible. Additionally, the information source, time of reception and reliability assessment is stored by each agent. These maps are updated through (1) personal observation (described next) and (2) communication with other agents (described in Section 3.4).

Personal Observation. The personal perception of an agent corresponds to its natural ability to observe its surroundings. Within a perceptible capability (e.g. visual and auditory range), an agent is considered to estimate the density around herself accurately. This personal observation acts as the *default* density perception of an agent of its current region, unless "outside" information is received, either through implicit dispersion due to sharing based on physical proximity communication, or explicit influence based on distant communication, in which case the decision model (Section 3.5) is used to update the agent perception map.

3.4 Communication

Physical Proximity Communication. Within an interaction range of radius R, all agents can interact with each other and share information about their own perception. The information exchange occurs, however, based on the decision model in Section 3.5 assuming a radius of range, $R = 25$ cells.

Distant Communication. Distant communication corresponds to the interaction between agents without spatial consideration, such as phone calls and messaging. For the simulations, we assume an agent attempts to contact another towards which she has maximum trust. Once the communication takes place, the perception maps *of both agents* would be updated based on the decision model (Section 3.5). It is possible the communication does not take place, if the receiving agent has already reached a PoA.

3.5 Decision Model

Very generally, an information source influences the confidence of an information receiver about the density in a region in proportion to the receivers trust to the source: the more the receiver trusts the source, the more it adopts the sources opinion on the density [19]. In emergency situations, people usually have little time and limited access to information to elaborate well possible decision options. Furthermore, available information is often contradictory, partial and outdated. Under these circumstances people often use cognitive shortcuts, such as based on trust. Given this reasoning, we formulate a decision model for our simulations.

The decision making of an agent consists of evaluating the time required for reaching each known exit. The agents estimation of the total time for each decision option (i.e., a path to an exit) depends on the agents estimation of its average speed for each sector on the path to the exit:

$$total\_time_{ag}(path) = \sum_{s \in path} \frac{l_{ag,s}(t)}{v_{ag,s}(t)} \tag{6}$$

The agents estimation of the length of the section $l_{ag,s}(t)$ of the path confined within sector s and of the average speed in the sector $v_{ag,s}(t)$ are updated based on the agents own observations and information about the crowd density in the sector received from other agents.

Information about the densities of regions are updated by decision making model. The higher the confidence value of the obtained information and the higher the trust of the agent to the agent-informer, the higher would be the effect of the obtained information on the agents beliefs:

$$B^*_{\rho_r,j} = \frac{C_{\rho_r,i} T_{j,i} B_{\rho_r,i} + C_{\rho_r,j} B_{\rho_r,j}}{C_{\rho_r,i} T_{j,i} + C_{\rho_r,j}} \tag{7}$$

where $T_{j,i}$ is j's trust towards i, and $*$ represents the value at the next iteration. Furthermore, $T_{j,i}$ is updated as:

$$T_{j,i}* = T_{j,i} + \alpha(C_{\rho_r,j} \frac{1}{1 + e^{-\gamma|B_{\rho_r,j} - B_{\rho_r,i}| + \beta}} - T_{j,i}) \tag{8}$$

where B is the belief, T is the trust, C is the confidence, and α, β, and γ are constants. We assume $C_{\rho_r,i}$ is the confidence agent i has about the about the density in region ρ_r. We assume an agent i communicates this density information to another agent j.

Agent js confidence is then updated as follows:

$$C^*_{\rho_r,j} = \frac{C_{\rho_r,i} T^*_{j,i} + C_{\rho_r,j} T^*_{j,j}}{T^*_{j,j} + T^*_{j,i}} \tag{9}$$

We assume every agent fully trusts themselves, therefore $T^*_{j,j} = 1$.

For simulation results, we assume $\alpha = 0.8$, $\beta = 5$ and $\gamma = 10$. Note, α is the rate of change of trust - a personality characteristic indicating the agents ability or willingness to change its state. β and γ are the steepness and threshold parameters of the logistic function, respectively. The values $\beta = 5$ and $\gamma = 10$ were chosen experimentally to reflect the following dynamics of trust:

- the agents gain high values of trust (> 0.7) slowly;
- a low level of trust (< 0.3) grows slowly with every positive experience;
- the average trust values ([0.3, 0.7]) vary rapidly.

Motivation for Trust. Agents associate trust with every relationship they have. Trust is an attitude of an agent towards an information source that determines the extent to which information received by the agent from the source influences agents belief(s). It takes values in the interval [0, 1]. The higher the trust to an agent, the higher the extent to which information provided by that agent is used in the decision making ([19]). The trust to a source builds up over time based on the agent's experience with the source. In particular, when the agent has a positive (negative) experience with the source, the agent's trust to the source increases (decreases). An information experience with a source is evaluated by comparing the information provided by the source with the agent's beliefs about the content of the information provided. The experience is evaluated as positive (negative), when the information provided by the source is confirmed by (disagree with) the agent's beliefs. This assumption is supported by many experimental evidences, which demonstrated that trust correlates positively with similarity of agents (e.g., similarity of interests) [20,21].

4 Experiments and Results

4.1 Simulation Scenario

For simulation results, we generate 5000 agents randomly, spread evenly over the walkable areas on the map (world). Therefore, each sector gets a fraction of agents equal to its walkable count over the total walkable area count. Our results are evaluated based on two scenarios.

1. **proximity comm**: The first evaluation of our models considers a scenario where communication only occurs based on physical proximity interactions. In this scenario there are no distant communications (defined in Section 3.4).
2. **full comm**: The second agent based model simulation considers a full communication model, where agent interactions occur both based on physical proximity as well as distant communications.

Of the many features simulated, we found the most critical to be the trust formations, the agent perception maps, as well as the distribution over exits, which is the focus of the results presented.

4.2 Trust Development

In order to evaluate the development of trust across the agents, we consider a network of trust. The nodes of the network are agents and the directed edges symbolize trust, where the weight of an edge is the amount of trust an agent has towards another. In order to understand the overall amount of trust in the

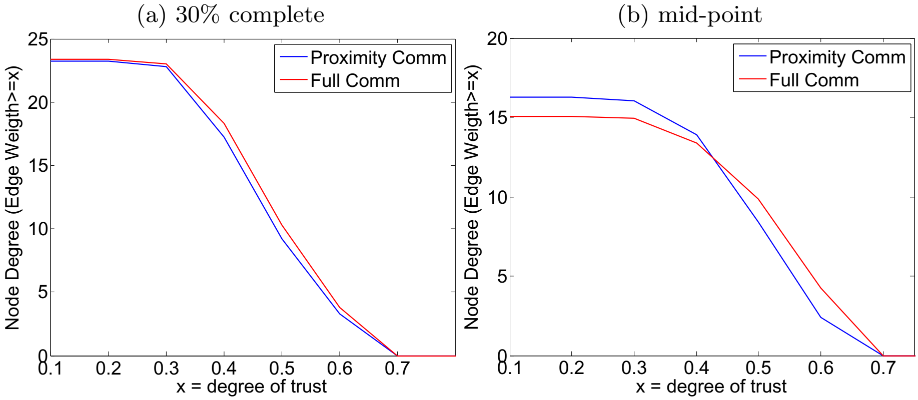


Fig. 2. The average agent node degree in the trust network plot as a function of degree (or amount) of trust. The network consists of agents as nodes and edges representing the degree of trust. In (a) the node degree is plot at the 15th iteration of the simulation (over a total of 50 simulations). In (b) the node degree is plot at the mid-point. In both cases, we can see there is a higher node degree for higher degrees of trust (indicating there is more trust in a full communication network). However, there is a higher node degree for lower degrees of trust given proximity only communication.

network, we consider the overall average node degree as a function of the degree of trust. More specifically, we consider the node degree for which the edge weight is greater or equal to x, as a function of x, where x is the degree of trust. These results are shown in Figure 2, where (a) is approximately the 1/3 point in time of the simulation and (b) is the mid-point of the simulation. We consider these points in time since these are the critical points at which agent interactions have taken place and the decision for PoA selection is vital at these instances. After the mid-point many of the agents reach their chosen PoA and therefore the simulation is stabile and the trust dynamics are no longer visible. Overall, the results in Figure 2 indicate that a full communication mechanism results in a higher number of pair-wise agent trusts given higher degrees of trust.

In Figure 3 we further make a comparison of (a) the overall distribution of the network of trust node degree from our agent based framework to (b) that of a real-life mobile phone data collection. The details of the mobile phone data collection and network data analysis can be found in [22]. In Figure 3 (b) we consider the overall static network of phone communications of the participants in the dataset and plot the average node degree as a function of the number of calling events (edge weight). Both plots in (a) and (b) are presented on a log-log scale. Very generally we observe their shapes to be similar, serving as a validity check for our trust model. We can conclude the network of trust developed by our agent based model generally follows a similar trend to a real phone communication network, where we assume phone communications occur between trusting individuals, with information exchange as is the case for the trust network.

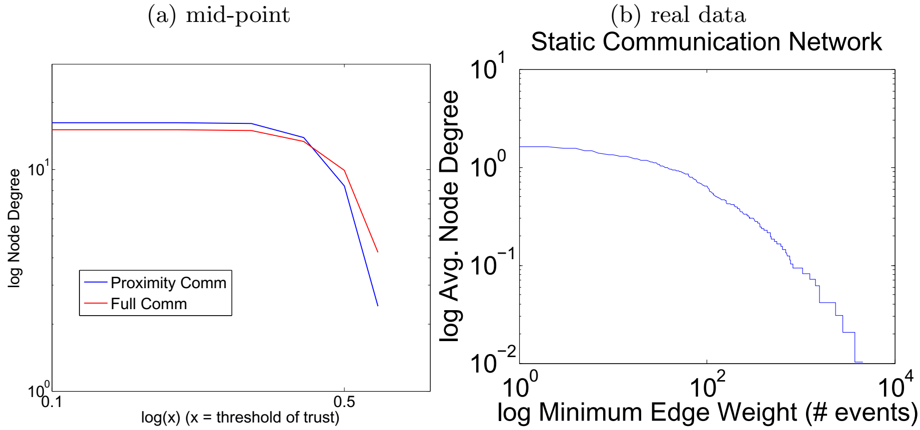


Fig. 3. Comparison of the average node degree as a function of edge weight for (a) the agent based model trust network at the mid-point of the simulation (b) the node degree of a real communication network taken from a large scale mobile phone data collection. This result serves as a general validity check to determine whether the network shape corresponds to that of a similar real-network.

4.3 Perception Maps

In order to evaluate the agent perception map data, we evaluate the *degree* to which each agent's perception of world is similar to the actual world information, which is the density of a sector. For this evaluation, we accumulate over all of the agents, the difference between the actual density and the perceived density ($|\rho_{actual} - \rho_{perceived}|$) for which the difference is greather than a threshold, Th. The results presented in Table 1 are computed over the total number of agents, iterations and sectors, resulting in the evaluation of the overall number of perception maps in the simulation.

Overall, we find that agents which communicate locally only are able have higher accuracy in their density perception of the world, though agents with a full communication mechanism, including both local and distant communications, are able to perceive more density information about the world (have more perception map information) though with slightly less accuracy.

4.4 Exiting Behavior

We plot the number of agents per point of attraction over time in Figure 4 to see how agents distribute themselves differenly in both simulation strategies. We observe that in the full communication simulation scenario, before the mid-point there is a more even distribution of agents to the PoAs (labeled as exits in the figure). This difference is subtle and can ben seen by the difference in the green curve between the figures at the mid-point.

Table 1. The number of perception maps with $|\rho_{actual} - \rho_{perceived}| < Th$, where the difference between the perceived density by an agent and the actual density is evaluated over different thresholds. The number of perception maps are computed over all the agents, sectors, and time steps. These results indicate that the full communication model results in overall more informed perception maps (as seen by $Th = 0.035$). However, when considering the least amount of error in the perceived density ($Th = 0.005$), the proximity only communication is more effective.

Th	(a) Proximity Communication $\lvert\rho_{actual} - \rho_{perceived}\rvert$	(b) Full Communication $\lvert\rho_{actual} - \rho_{perceived}\rvert$	(b)-(a) $difference$
0.005	3764170	3742735	-21435
0.01	4219515	4206839	-12676
0.02	4903318	4907265	3947
0.03	5128701	5133970	5269
0.035	5228494	5239598	11104

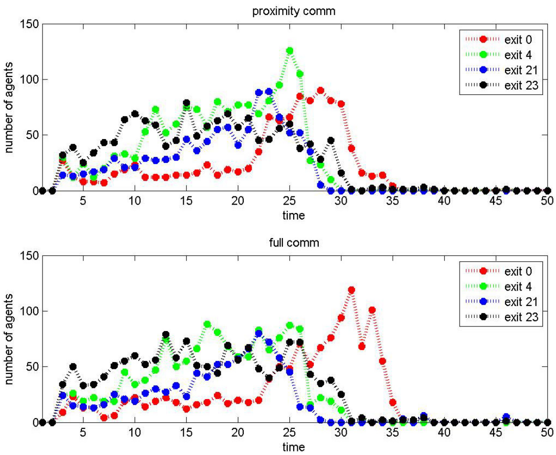


Fig. 4. The number of agents having reached a PoA (labeled as exit x) over time. In the full communication scenario, before the mid-point, the four points of attraction are more evenly reached. This can be seen by the spreading of agents from the PoA in green (exit 4) to other PoAs.

5 Conclusion

We present an agent based simulation framework to model the dynamics of agent perception and explore the effect of communication on crowd dynamics in the context of evacuation. We present a new model defining a belief, confidence and trust mechanism which forms the basis for agent movement decision making based on the agent density perception map. We have found a full communication model to be advantageous to a local communication model since agents can have a larger overall number of agent perceptions about the world, and can result in a larger number of highly trusted pairwise relationships.

References

1. Lazer, D., Pentland, A., Adamic, L., Aral, S., Barabási, A.L., Brewer, D., Christakis, N., Contractor, N., Fowler, J., Gutmann, M., Jebara, T., King, G., Macy, M., Roy, D., Van Alstyne, M.: Computational social science. Science 323(5915), 721–723 (2009)

2. Sun, R.: Cognition and multi-agent interaction: From cognitive modeling to social simulation. Cambridge University Press (2008)
3. Alam, S., Geller, A.: Networks in agent-based social simulation. Springer (2012)
4. Kleinberg, J.: The small-world phenomenon: an algorithm perspective. In: ACM STOC, pp. 163–170. ACM, New York (2000)
5. Liben-Nowell, D., Novak, J., Kumar, R., Raghavan, P., Tomkins, A.: Geographic routing in social networks. PNAS 102(33), 11623–11628 (2005)
6. Lambiotte, R., Blondel, V., Dekerchove, C., Huens, E., Prieur, C., Smoreda, Z., Vandooren, P.: Geographical dispersal of mobile communication networks. Physica A: Statistical Mechanics and its Applications 387(21), 5317–5325 (2008)
7. Krings, G., Calabrese, F., Ratti, C., Blondel, V.D.: Urban gravity: a model for inter-city telecommunication flows. Journal of Statistical Mechanics: Theory and Experiment 2009(07), L07003+ (2009)
8. Adamic, L.A., Adar, E.: How to search a social network. Social Networks 27 (2005)
9. Mok, D., Wellman, B., Basu, R.: Did distance matter before the internet? interpersonal contact and support in the 1970s. Social Networks 29, 430–461 (2007)
10. Helbing, D., Farkas, I., Vicsek, T.: Simulating dynamical features of escape panic. Nature 407, 487–490 (2000)
11. Massaguer, D., Balasubramanian, V., Mehrotra, S., Venkatasubramanian, N.: Multi-agent simulation of disaster response. In: ATDM Workshop in AAMAS, vol. 2006. Citeseer (2006)
12. Murakami, Y., Minami, K., Kawasoe, T., Ishida, T.: Multi-agent simulation for crisis management. IEEE KMN, 135–139 (2002)
13. Jain, S., McLean, C.: Simulation for emergency response: a framework for modeling and simulation for emergency response. In: Proc. of the 35th Conference on Winter Simulation: Driving Innovation, Winter Simulation Conference, pp. 1068–1076 (2003)
14. Bosse, T., Hoogendoorn, M., Klein, M.C.A., Treur, J., Wal, C.N., Wissen, A.: Modelling collective decision making in groups and crowds: Integrating social contagion and interacting emotions, beliefs and intentions. In: Autonomous Agents and Multi-Agent Systems, pp. 1–33 (June 2012)
15. Tsai, J., Bowring, E., Marsella, S., Tambe, M.: Empirical evaluation of computational emotional contagion models. In: Vilhjálmsson, H.H., Kopp, S., Marsella, S., Thórisson, K.R. (eds.) IVA 2011. LNCS, vol. 6895, pp. 384–397. Springer, Heidelberg (2011)
16. Sharpanskykh, A., Zia, K.: Emotional decision making in large crowds. In: Demazeau, Y., Müller, J.P. (eds.) Advances on PAAMS. AISC, vol. 155, pp. 191–200. Springer, Heidelberg (2012)
17. Collier, N.: Repast HPC Manual. Technical report, p. 44 (November 2010)
18. Kirchner, A., Schadschneider, A.: Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. Physica A: Statistical Mechanics and its Applications 312(1), 260–276 (2002)
19. Golbeck, J.: Computing and Applying Trust in Web-based Social Networks. PhD thesis, University of Maryland, College Park, College Park, MD, USA (2005)
20. Ziegler, C.N., Golbeck, J.: Investigating correlations of trust and interest similarity. Decision Support Systems 43(2), 460–475 (2007)
21. Bhuiyan, T.: A survey on the relationship between trust and interest similarity in online social networks. JETWI 2(4), 291–299 (2010)
22. Farrahi, K., Emonet, R., Ferscha, A.: Socio-technical network analysis from wearable interactions. In: ISWC, Newcastle, UK (June 2012)

SerenA: A Multi-site Pervasive Agent Environment That Supports Serendipitous Discovery in Research

Jamie Forth[1], Thanasis Giannimaras[1], Geraint A. Wiggins[1], Robert Stewart[2], Diana Bental[2], Ruth Aylett[2], Deborah Maxwell[3], Hadi Mehrpouya[3], Jamie Shek[3], and Mel Woods[3]

[1] School of Electronic Engineering and Computer Science
Queen Mary University of London, Mile End Road, London, England, E1 4NS
[2] School of Mathematical and Computer Sciences
Heriot-Watt University, Edinburgh, Scotland, EH14 4AS
[3] Duncan of Jordanstone College of Art and Design
University of Dundee, Perth Road, Dundee, Scotland, DD1 4HT

Abstract. We present SerenA, a multi-site, pervasive, agent environment that suppers serendipitous discovery in research. The project starts from the premise that human users cannot be aware of all the research information that is relevant to their work, because of the compartmentalisation of research into fields around particular journals, and, simply, because there is too much to know. In particular, the Semantic Web provides a resource which can assist, but there is more to be discovered than the things that a user might deliberately search for. SerenA, then, attempts to assist researchers by presenting them with information that they did not know they needed to know about their research.

Keywords: Serendipity, semantic web, pervasive agents.

1 Introduction

We describe SerenA, an agent-based, semantic, pervasive, embedded personal assistant system, for academic researchers. SerenA is designed in response to the RCUK Digital Economy "Designing Effective Research Spaces" challenge[1], and is, in the first instance, intended for arts and humanities researchers.

The key idea of SerenA is to create a Serendipity Arena: a virtual space in which serendipitous discovery of several different kinds is more likely to happen than elsewhere. For example, two SerenA users with common research interests might be travelling to the same place, but not be aware of the fact; SerenA would alert them and, if its users desired, arrange a discussion meeting. However, the aim is not merely to create an academic dating service; rather, the idea extends to objects in the world too, so that, for example, a researcher arriving at King's Cross station in London could be informed that a unique manuscript,

[1] `www.epsrc.ac.uk/SiteCollectionDocuments/Calls/2009/DERSSandpitCall.pdf`

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 85–96, 2013.

of direct interest to their work, and borrowed temporarily from Egypt, is on show in a nearby museum on that day; what is more, they could be told this in advance, when they book their train ticket on line, so they can plan for a visit. Above all, SerenA is meant to find for its users things that they did not know they wanted.

These superficially simple tasks involve significant background reasoning, about users' interests and activities, and about real-world data, such as location and time, both current and planned. They require a substantial software infrastructure that can travel with the user, but also supply substantially more computational power than a standard mobile device; this is sometimes referred to as "the Mobile Dilemma". The infrastructure must also be capable of semantic analysis of a user's writings and actions, if not in real time, at least in a time frame that is practically useful in context (consider the examples above: the temporal requirements of the corresponding notifications are quite different). SerenA's target users cannot be assumed to be expert computer users, and therefore it must work with them in ways that are both easy to use and engaging.

Thus, under detailed scrutiny, the superficially simple idea of SerenA quickly expands from an engaging and useful tool to an archetypal general AI problem, including language understanding, proactive semantic reasoning, intelligent interaction and pervasive presence. The only factor to reduce the challenge is the nature of serendipitous discovery itself: unusually in computer science, and only within reason, the system can be *usefully ambiguous or even wrong*. This is so because, ultimately, SerenA forms a hybrid system with its users, and unexpected, off-the-wall information from the computational part of the hybrid can nevertheless usefully inform or stimulate the human part. Therefore, SerenA has reduced responsibility for correct reasoning: its suggestions need only be useful; they do not have to be logically correct.

Luck et al. [5, p. 40] identify seven key challenges for multi-agent system research, all of which are relevant to SerenA—though not all are currently implemented. Here, we exemplify challenges 4.2 (increase quality of agent software to industrial standard), 4.3 (provide effective agreed standards to allow open systems development), 4.4 (provide semantic infrastructure for open agent communities) and 4.6 (develop agent ability to understand user requirements). The remaining challenges are for future work. This said, the key contribution described in this paper and associated demo, on the practical applications of agents, is the *assemblage* of the system, not its components.

The funded SerenA project, as a whole, includes designers, HCI and usability specialists, and computer scientists, working together in a broad coalition. It has considered the epistemology of serendipity [8,6,7], the meaning of the concept of serendipity to researchers, and their reactions to it [13], and approaches to concept extraction from text [11]. Also work has been done on language processing with a view to discovering users' interests from their tweets [12] and their goals from their notes and email messages [10]. The current paper, though, takes a top-down perspective, explaining the overall conceptualisation of the system, explaining how the primary challenges are met, and in particular focusing on the agent system that forms the core of the pervasive environment.

With this aim in mind, the rest of the paper describes a selection of the functions of SERENA, from a user's perspective, identifying the engineering challenges, and outlines how they are integrated by SERENAś core agent system. There is not space here to describe every aspect of SERENA in detail. Rather, we aim to give a flavour its capabilities, and to demonstrate how the agent system contributes, on multiple levels, to the elegant implementation of the whole.

2 Affordances and Constraints

We now describe SERENA from the user's point of view, explaining how the various affordances of the software affect its design. We identify the points at which the agent approach is particularly useful.

2.1 Ubiquitous Intelligent Personal Support

SERENA is conceived as a ubiquitous computing environment, which should interact with its users via their mobile devices and desktop computers, and via public installations in public spaces. The requirements of these three *interactors* are somewhat different. The first two are *private* to the user, and authenticated, while the third is *public*: this difference impinges on the nature of the information that can be displayed. Mobile devices tend to have significantly smaller displays than desktop machines, and a public display is more likely to be large than small, so this dimension distinguishes the first category of interactor from the second and third. In the SERENA project so far, we have focused on the mobile private SERENA, with only one proof-of-concept public SERENA interactor currently planned. These are described in §2.2.

SERENA's ubiquity, delivered through mobile and fixed devices, imposes constraints on its design, which agent-based designs are well suited to meet. First, the behaviour of the system must be consistent and persistent. That is to say, an interactor must behave uniformly, and must not lose information if, for example, it is switched off, or (more likely) if there is a network outage. A very neat way of addressing this issue is to maintain an agent that simulates the mobile device, and then to implement synchronisation of information between the mobile device and its agent, which can be done independently of the workings of the rest of the agent system, and of the user's interaction with the device[2]. We term this kind of agent, that echoes the behaviour of an entity in the physical world, a *shadow* agent[3]. Shadow agents of other kinds appear too, the most important being the user agency, a group of agents that shadow the user, supplying information about the user (e.g., location, research interests, privacy settings) to

[2] Of course, the user's interaction with the device as a whole may be restricted by a temporary network outage, but this is not a soluble problem.

[3] The term is borrowed from the UK government system, where the official Opposition forms its own cabinet, *shadowing* the ministerial (dys)functions of the government.

the rest of the system, but also autonomously acting on their behalf to make serendipitous discoveries. The user agency is described in §3.2.

A key advantage of the shadowing approach is that it allows us to meet SerenA's requirements for high-power computing and network access in static installations with high-powered servers with fast access to the Internet, both of which are *sine qua non* for the deep inference required of SerenA if it is to be helpful to its users. A further concomitant advantage is the stability and security of such managed systems. The agent community, running on these servers, can work uninterrupted, communicating internally with the shadows of the physical world interfaces, and those interfaces can be updated live when connected, or asynchronously when an interactor reappears on the network after being disconnected. What is more, the asynchronous nature of the agent community absolves SerenA's implementers of the need to manage the appearance and disappearance of interactors, and also the delays inherent in web services and sites; agents communicate via message passing, and, simply, when they are disconnected, or when there is a delay, no external messages messages appear (though of course the corresponding shadow agent may continue working independently on the basis of prior information). A final bonus of shadowing is in the use of shadow agents to represent external internet resources within the agent community. This means that issues of translating between languages and formats need not be spread throughout the system, but can be handled by a specialist shadow, so that the information is manipulated into a SerenA-friendly form, exactly once, as soon as it enters the system. Our use of FIPA ACL standards[4] means that responses can be straightforwardly associated with their corresponding queries, on receipt, using conversation management.

SerenA is implemented using the Java Agent DEvelopment Framework[5] [1], which supports two important SerenA requirements. In JADE, agents run in notional *containers*, one or more containers per machine, but multiple machines can be connected together into a JADE *platform*. This allows failover to be implemented at the agent level: duplicate containers of SerenA agents can be run at multiple sites, and one fails, another can take over.

2.2 Interactors

Private SerenA

One of SerenA's design constraints is that its users should not have to use specialist interfaces and equipment; it should work invisibly behind familiar interaction paradigms, making itself noticed only when a direct user response is required, or when the user's attention needs to be drawn to SerenA output. To this end, we have designed an interface, which is the current focus of our interaction work, as an Android app, conceptualised as a *Semantic*[6] *Notebook*

[4] `http://www.fipa.org/specs/fipa00061/SC00061G.html`

[5] JADE; `http://jade.tilab.com`. It seems likely that Erlang (`www.erlang.org`) is a future candidate for such implementations.

[6] This epithet is justified further in §3.

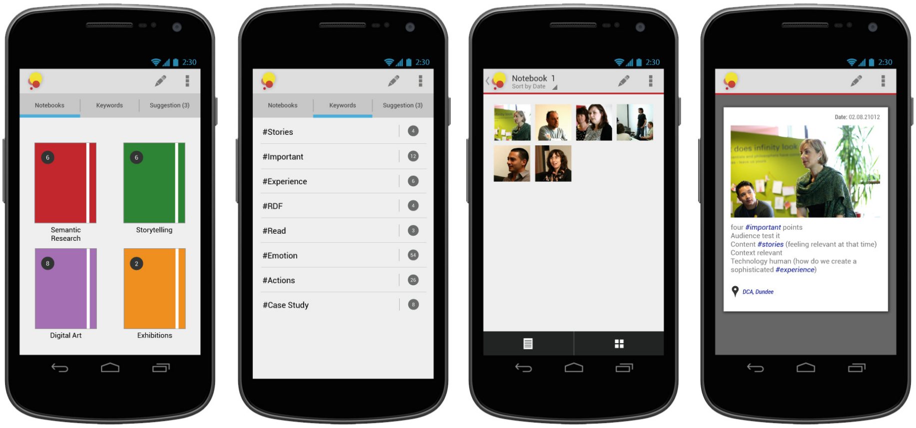


Fig. 1. The Semantic Notebook App. Left to right: Notebook view (creating a new notebook and personalising the cover); Goal list (keywords and goals); Sort By: Visual and date; Keywords and Multimodal notes with Goals (Tags in the body of the text)

[9]. The researcher-user is invited to make notes, add tagged images, keywords, and so on, all in free text. The notes can be organised in a predictable but useful way. Interaction with the user is then managed by SERENA processing the user's text, and then adding annotations (e.g., items of text, web links), to the notes, making the distinction between the user's notes and SERENA's additions clear by means of typography. Example views are given in Figure 1. When SERENA adds a new annotation, the Android notification system informs the user, according to their preferences. The user can then follow up the suggestions at their leisure, or delete them if they wish. In some cases, time-sensitive notifications will fade away when they become stale, though a user might choose to override this.

This kind of interaction is advantageous for two reasons: first, SERENA must avoid the paperclip effect[7]; and, second, the reasoning required to suppress pointless information is often extensive (see §3), and cannot be performed on the fly, while the user waits. The concomitant advantage of asynchrony is that network outages do not degrade the experience: the user will come to expect SERENA suggestions at some time after they make their notes, but not immediately.

Public SERENA
The criteria for a public instantiation of SERENA are quite different. First, there are significant security issues concerned with a user's personal or private information: in the current prototype, we address this simply by using information that we know to be public. There are major open opportunities in a public installation of this kind: it is not merely intended to be a terminal that individuals

[7] The irritation produced when Microsoft's Clippy character used to intrude unexpectedly, distracting the user from their task, with often incorrect information.

can use to access private SerenA, but something altogether more collective. One key affordance, given the right social context, is the ability to communicate with more than one user at once via the same channel. For example, at an academic conference, one might display connections between consenting delegates, in terms of the connections that SerenA has found between their work and interests: since SerenA is focused on finding *unexpected* connections, this approach might be expected to add value to the social interaction at the event.

Our first prototype public SerenA is conceived in the context of a major UK city library. It is architecturally melded with the building, in that its outputs are projected directly on to walls, using site-specific designs that integrate with the architecture. Its outputs consist of simple visualisations (the simplest being mere text) of documents that are ordered via the library's on-line order system. The information is filtered, so that no connection with the library user ordering it can be made: their name is not displayed, nor is the time at which they ordered it. In principle, however, interesting work could be done finding connections between documents ordered by different people, and displaying on public SerenA a summary of these: a sort of local, temporary Zeitgeist analysis. An example design is shown in Figure 2.

Communication with the Agent System

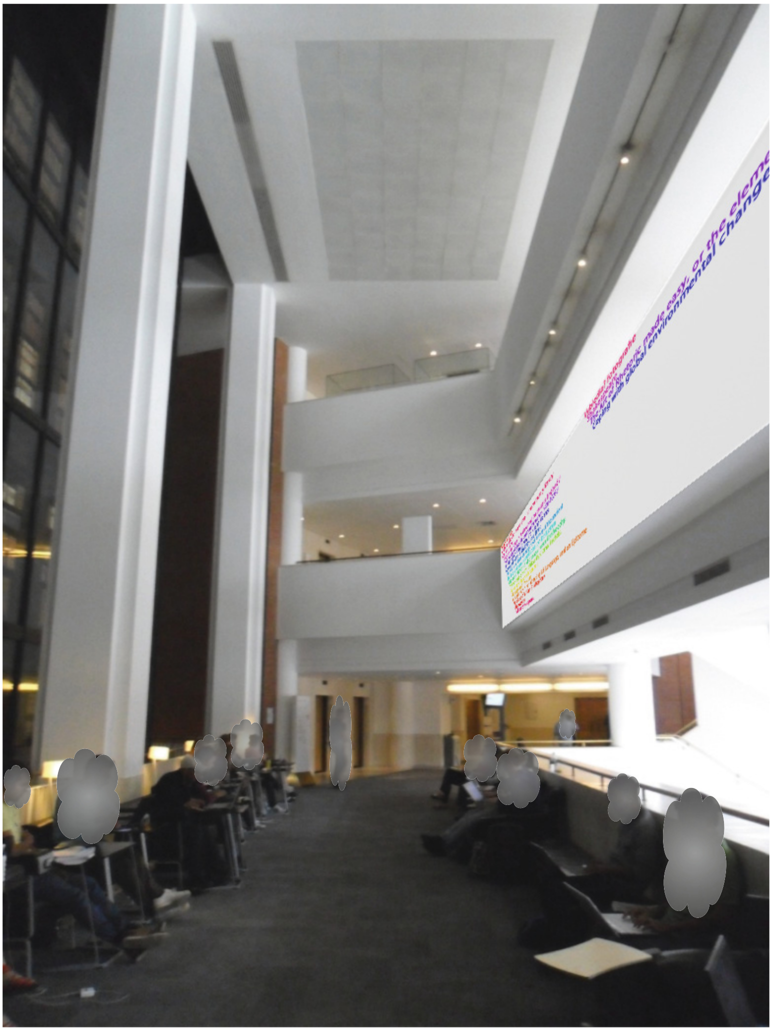
SerenA's interactors communicate with the Agent System via shadow agents as outlined in §2.1. Externally, the connection from the shadow to the mobile device is implemented directly in Java, running as a background process in Android, and connecting to the shadow via a persistent WebSocket[8,9]. This process then communicates with front end UI managers, such as the Semantic Notebook App. Necessarily, some local storage of information (such as what is on the current display, what is in the notebook, both from user and from SerenA, and what has or has not been synchronised with the shadow agent) must be managed, and this is done using a local database, working in the same language as the agent system (see §3); other issues of synchrony are dealt with at the level of WebSockets or below, in TCP/IP.

3 Knowledge Representation and Inference to Support Serendipity

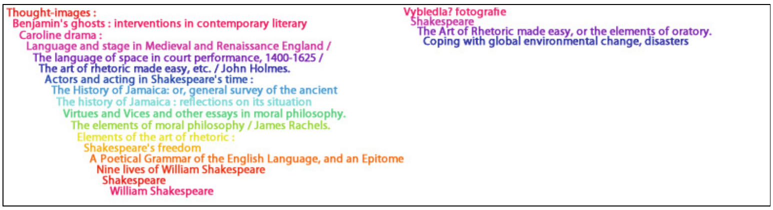
3.1 Formalism: The Semantic Web

In order to make interesting and unexpected connections, SerenA explores and combines information from multiple sources, using the growing array of Semantic Web resources. Increasing amounts of information from many different domains are being made available as Linked Open Data (LOD). LOD uses syntactic and semantic standards such as The Resource Description Framework[10] (RDF) and

[8] http://www.w3.org/TR/websockets/
[9] http://tools.ietf.org/html/rfc6455
[10] http://www.w3.org/RDF/



a)



b)

Fig. 2. An example design for one instantiation of public SerenA. Books being ordered from a library are presented as part of the architectural structure of the library in near-real time. a) The live display. b) Direct view of the information displayed.

OWL[11] (the Web Ontology Language), and is available for query across the web. SERENA also uses Semantic Web ontologies currently being developed and integrated to express information in specific domains, such as FOAF[12] to describe people and relationships, DBpedia[13] for general knowledge, GeoNames[14] for geographic locations and DBLP[15] and Dublin Core[16] for publications. These ontologies and resources are being realised in individual web-accessible databases which can be searched by tools such as Sindice[17] or merged into larger databases of machine-readable information such as FactForge[18]. As a result of these initiatives it is now possible to combine information from many different sources, at a general, domain-independent level, to link location data obtained from a mobile device to information about nearby places of interest, for example.

SERENA's domain knowledge and agent control commands are represented in RDF, a good choice for expressing highly structured knowledge-based information; together with the OWL-DL subset of OWL, this affords a Description Logic [2, ch. 9], a well-understood basis for knowledge representation and inference. SERENA's agents communicate in RDF, also, and its message envelopes meet the FIPA Agent Communication Language specification. This approach eliminates the need to translate domain knowledge acquired from the Semantic Web to internal agent knowledge representations. It can also support multiple levels of agent reasoning.

Because SERENA agents are themselves described in RDF, the possibility is open to build agents that reflect on and modify the behaviour of others. However, this is currently deferred to future work.

3.2 User Modelling: Agent.Me

Over time, SERENA builds a model of its user, expressed in RDF, including information given by the user, information inferred directly under the control of the user, and information inferred about the user by the system. This information is made available to a collection of agents, which act as the user's shadow in the agent community. The modelling process is kick-started by our Discover.Me.Semantically service[19], which searches for information about a new user, and then consults with them to select what is relevant.

[11] `http://www.w3.org/TR/owl2-overview/`
[12] `http://www.foaf-project.org`
[13] `http://dbpedia.org`
[14] `http://www.geonames.org`
[15] `http://dblp.uni-trier.de`
[16] `http://dublincore.org/documents/dc-rdf/index.shtml`
[17] `http://sindice.com`
[18] `factforge.net`
[19] Source code under GPLv3 license:
`https://github.com/robstewart57/discover-me-semantically`
Running instance:
`http://serena.macs.hw.ac.uk/serena/discover-me-semantically/`

Discover.Me.Semantically is a web-based tool that allows its user to author RDF representing their professional and personal interests, skills and expertise. The stand-alone implementation allows the user to download this RDF representation as a file to be hosted on their own web pages[20].

The standalone implementation (which the reader is invited to try) also offers a path to visualize this RDF on a linked-open-data visualization called LodLive[21], to explore the paths along links away from their skills and interests. The tool knows of several Web resources, and records some aspects of equivalence between them, and these *sameAs* links can also be explored. The RDF representation of the user so generated can also be stored in SERENA's user model, and then be used by a user's shadow, or Agent.Me to assist in finding information of interest. Other readily available information sources also contribute to the user model, such as bibliographical information, obtained from a user's BibTeX or EndNote file. In the longer term, we intend to repurpose ideas from Intelligent Tutoring Systems, in which the computer's model of a user is made visible to the user [3], so that the user can reflect on it, and correct it if necessary.

3.3 Supporting Serendipity with Inference: Goal Detection

It is in the nature of serendipity to be unpredictable: if one could create the effect to order, it would not be serendipity, by definition. Our aim, instead, is to enhance the conditions in which serendipity might take place. Ultimately, this process will be managed by the agent system, with some agents requesting information to give to the user, and others searching for answers according to the competence of the resource they are shadowing. The asynchronous, open nature of the agent system, and also of our interface method (see §2.2) mean that relatively little overhead needs to be expended on simple question-answer interaction. More interesting reasoning, however, can be carried out by generalised reasoning agents, but this is future work.

A key issue in supplying the user with useful information is to understand the research goals that they are expressing in their notes, files and email[22]. We have begun work with the GATE natural language processing system[23], with some success in detecting goals in natural language sentences [4], and an ontology for goals has been defined [10]. These ideas will inform the Agent.Me.

An example of these ideas in action can be found in our case study of connecting users within one institution. For this, we use information gathered for the UK Research Excellence Framework[24]. The information is about publications of every academic in the institution (coverage is not universal, in fact, but this does not prevent the system from working), and about the academics themselves.

[20] Having a `foaf.rdf` file attached to one's academic webpage is becoming commonplace for Semantic Web based researchers.

[21] `http://lodlive.it/`

[22] Of course, SERENA does not access files or email without permission.

[23] `http://gate.ac.uk`

[24] `http://www.ref.ac.uk/`

This information is available in RDF format, delivered via a web service. Also included, for many publications, is the plain text of the abstract.

Our approach is to extract semantic annotations from titles and abstracts, using the OpenCalais web service[25]. Once this is done, we run semantic web queries to deduce answers to such leading questions as:

- Which people in different schools (who therefore may not know each other) describe the same concepts in their papers?
- Are there more experienced specialists (e.g., professors) who often publish papers on concepts also of interest to (e.g.) early-career research associates?

Perhaps unsurprisingly, these queries produce many answers, and we are working on heuristics to filter them in a useful way, with respect to promoting serendipity.

4 System Overview and Example

Figure 3 illustrates the overall system architecture with some example agent types, and also shows the potential communication between agents in two specific tasks. These are: information flow resulting from the addition of a new note to the Semantic Notebook App (dashed arrows); and information flow resulting from the arrival of two users with common interests in a location with a public SerenA installation (dotted arrows).

New Notebook Annotations
When User 1's device shadow agent announces to User 1's Agent.Me that a new note has been added to the user's Semantic Notebook, the associated information seeker agent broadcasts a request for information on the tags included in the note. Some time later, a web resource agent finds a semantic web resource that referring to some of the same concepts. The resource is returned to the user's interactor via its shadow, and presented in the Notebook as a title with a clickable link.

The same broadcast request also reaches an inference agent, whose speciality is to make connections between SerenA users who publish on related concepts in different research fields. This agent can search for publications in DBLP, by concept; it then broadcasts a request to all Agent.Mes to ask for the research fields and institutions of SerenA's users whose papers it finds. It then returns an answer to User 1's Agent.Me listing the papers of those users in their institution, who publish on the concepts of interest but in different research fields.

Arrivals and Meetings
A more complicated example arises when two SerenA users who have enabled their location agents arrive at the same conference, where there is a public SerenA interactor. The users' location agents broadcast their arrival at the

[25] `http://www.opencalais.com`

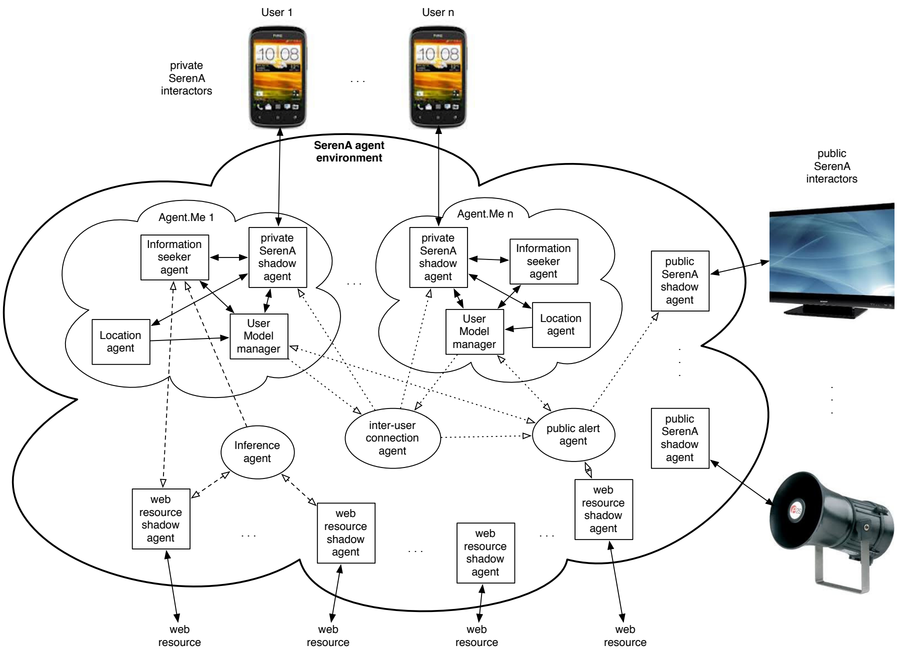


Fig. 3. Overview of SERENA, with two potential agent conversations. Solid arrows indicated on-going administrative information flow. Dashed arrows show communication resulting from the addition of a new note to the Semantic Notebook App on User 1's phone. Dotted arrows show communication resulting from the arrival of two users with common interests in a location with a public SERENA installation. See §4 for details.

event (which their Agent.Mes infer is significant from their diary entries), and the inter-user connection agent notices their physical coincidence. It queries their respective user models, and learns that they are both interested in being introduced to other SERENA users. The inter-user connection agent broadcasts the opportunity to meet, which is relayed to the users' interactors by their Agent.Mes, but is also picked up by the local public alert agent, associated with the public SERENA installation. This agent checks with both users that they allow their images to be used on public SERENA, and, if they do, tells the shadow agent to announce their presence and introduce them to each other.

5 Conclusion and Future Work

In a paper of this length, it is not possible to cover all the aspects of a system design as large and diverse as that of SERENA. Here, we have attempted to convey the basic ideas and *raisons d'être* of the system, and to give enough detail to explain the contribution of the agent framework and communication style, and the resulting interaction with users. We believe this to be a novel contribution to practical applications of multi-agent technology, in that to our knowledge, a distributed agent system of this scale has not previously been deployed.

There is, it is clear, a substantial amount of work to do before we can claim that SerenA has fulfilled its potential—although formal evaluation of various aspects of the work outlined in this paper is in progress, there are many more possibilities for emergent behaviour that have yet to be enabled. Detecting and enhancing such behaviours will be the focus of future work.

Acknowledgements. This work is supported by EPSRC Sandpit Research Grant EP/H042741/1 – SerenA: Chance encounters in the Space of Ideas.

References

1. Bellifemine, F., Bergenti, F., Caire, G., Poggi, A.: Jade — a Java agent development framework. In: Bordini, R., Dastani, M., Dix, J., Fallah Seghrouchni, A. (eds.) Multi-Agent Programming. Multiagent Systems, Artificial Societies, and Simulated Organizations, vol. 15, pp. 125–147. Springer, US (2005)
2. Brachman, R.J., Levesque, H.J. (eds.): Readings in Knowledge Representation. Morgan Kaufmann (1985)
3. Bull, S., Dimitrova, V., McCalla, G.: Open learner models: Research questions. International Journal of Artificial Intelligence in Education 17(2), 83–87 (2007)
4. Giannimaras, A.: Goal detection in natural language text. MSc thesis, School of Electronic Engineering and Computer Science, Queen Mary University of London (2012)
5. Luck, M., McBurney, P., Preist, C.: Agent Technology: Enabling Next Generation Computing. Agentlink (2003)
6. Makri, S., Blandford, A.: Coming across information serendipitously: Part 1: A process model. Journal of Documentation 68(5), 684–705 (2012)
7. Makri, S., Blandford, A.: Coming across information serendipitously: Part 2: A categorisation framework. Journal of Documentation 68(5), 706–724 (2012)
8. Makri, S., Erdelez, S.: Thematic issue on opportunistic discovery of information. Information Research 16(3) (2011)
9. Maxwell, D., Woods, M., Makri, S., Bental, D., Kefalidou, G., Sharples, S.: Designing a semantic sketchbook to create opportunities for serendipity. In: Proceedings of BCS HCI Conference (2012)
10. Piao, S., Bental, D., Whittle, J., Aylett, R., Makri, S., Sun, X.: A pilot study: Deriving a users' goal framework from a corpus of interviews and diaries. In: Proceedings of the Workshop on Semantic Relations II - Enhancing Resources and Applications (2012)
11. Piao, S., Forth, J., Gacitua, R., Whittle, J., Wiggins, G.: Evaluating tools for automatic concept extraction: A case study from the musicology domain. In: Proceedings of Digital Futures 2010 (2010)
12. Piao, S., Whittle, J.: A feasibility study on extracting twitter users' interests using nlp tools for serendipitous connections. In: Proceedings of the Third IEEE International Conference on Social Computing (SocialCom 2011). IEEE (2011)
13. Sun, X., Sharples, S., Makri, S.: A user-centred mobile diary study approach to understanding serendipity in information research. Information by Chance: Opportunistic Discovery of Information 16(3) (2011)

Towards a Multi-avatar Macroeconomic System

Gianfranco Giulioni[1], Edgardo Bucciarelli[1], Marcello Silvestri[2], and Paola D'Orazio[1]

[1] Department of Economical-Statistical and Philosophical-Educational Sciences, University of Chieti-Pescara, Italy
[2] Confindustria Research Unit, Pescara, Italy

Abstract. In this paper we provide a prototype Multi-Agent macroeconomic system where artificial agents' behavioral rules are calibrated using human behaviors. The artificial representation of humans, i.e. "molded" avatars, derives from a two steps process. The first step consists in performing laboratory experiments in order to gather microeconomic data. The second relies on the use of data mining heuristic techniques on the experimental data which allow for the identification of the behavioral models that better approximate humans' behaviors. The implementation of these techniques, which we call *microeconomic calibration*, leads to the identification of heterogeneous agents. We maintain that the Multi-Avatar systems which arise from the heterogeneous microeconomic calibrated agents give the possibility to improve the existing Agent-based models. The proposed approach will generate consequently more reliable policies evaluation tools. The model discussed in the paper is motivated by the investigation on the role of entrepreneurs' financial decisions over the business cycle. This topic was addressed especially by Hyman Minsky since the Fifties and recently gained a new momentum in the scientific community after the onset of the financial crisis.

Keywords: Macroeconomics, Experimental Economics, Agent-based Computational Economics, Microeconomic Calibration, Financial Instability Hypothesis.

1 Introduction

Recently, a growing number of economists has been showing an increasing interest in the application of Multi-agent Systems methodologies, in particular in the building of Agent-based models. The increasing "enthusiasm" in the scientific community for the potentialities of this approach - especially for understanding macroeconomic dynamics and policies [2,12] - gave birth to a new strand of Economics which is called Agent-based Computational Economics (ACE) [24].

Among alternative modeling tools, ACE has proven to be valuable since it accounts for "realism": market disequilibria, bounded rationality of agents and heterogeneity. ACE allows for modeling a large range of economic behaviors and solves the aggregation problem by summing up over all individual variables.

Nevertheless, ACE is not free from criticism. *Ceteris paribus*, one of the major challenges for researchers is to give scientific justification to the structure of interaction and to agents' behaviors selected and included into the model.

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 97–109, 2013.

In the last few years the literature on ACE modeling has blossomed, but in our opinion there are opportunities to strengthen the topic of how to endow artificial agents with the "correct" behavior. Researchers can indeed choose among several options[1] in order to model agents' actions, but this brings the "(too) many degrees of freedom" problem into ACE modeling (for a broader view on this issue see for example [11]).

In this paper we maintain that the construction of a reliable macroeconomic Agent-based model should implement a process based on the following steps:

1. Inducting and calibrating microeconomic behavioral models by using "real world" microeconomic data;
2. Building an artificial agent for each identified behavioral model;
3. Building an Agent-based model using the artificial agents designed in the previous step;
4. Validating and calibrating the macroeconomic outcome of the artificial economy by using "real world" macroeconomic data.

This process has the advantage of performing a double validation-calibration exercise (both at the micro and macro level) which in turn increases the reliability of the model for policies evaluation, thus improving the guidance of policy making applied to different frameworks.

However, the gathering of relevant microeconomic data is one of the first problems which arises from the implementation of the process presented above. In our work we try to solve this problem by running economic experiments.[2] In the present paper we briefly report on the experiments we carried on[3] while we focus on how to induce microeconomic behavioral rules and how to use them to build an Agent-Based Model. These phases of our investigation are described in sections 2 and 3 respectively.

We maintain that the suggested new methodology could address the issue of microfoundations[4] with more details than the existing literature. We suggest that

[1] In the literature there have been developing two main approaches. The first relies on Behavioral Economics theories and methods: it consists in the shaping of individual behaviors through psychological experiments on human subjects (for an introduction to the topic see [1] and [3]). A second approach is complementary to the first one: it suggests to select only those behavioral rules which reproduce *stylized facts* at aggregate level (see e.g. [8] and [6]).

[2] For a comprehensive survey on the use of economic experiments in Macroeconomics see [10]. One of the first economists who called the attention on the complementarity of experimental economics and ACE is John Duffy who argued that bottom-up, boundedly rational, inductive models of behaviors provide a better fit to experimental and field data. For a broader assessment of this topic see [9].

[3] For further analysis concerning the model on which the presented paper is rooted on see [16,17].

[4] The microfoundations of traditional macroeconomic models (DSGE models) are still a point of contention among economists [5]. The "crosstalk" between Solow [22] and Chari and Kehoe[4] is of particular interest for understanding the core issues under debate.

inducting and calibrating behavioral rules by using experiments could provide a deeper analysis of the individual decision making.

The underlying economic model is inspired by Hyman Minsky's thought [19] and in particular by his well known Financial Instability Hypothesis. Minsky's ideas have been widely neglected by the economic profession since his main concepts are very difficult to model using the traditional *deductive* analytical approach[5]. In contrast to that criticism we argue that the *inductive* approach, i.e. finding entrepreneurs' financial behavior by using experiments, is a valid way out of that methodological *impasse*.

2 Identifying Microeconomic Behavioral Models

In this section we firstly describe the experimental settings which are intended to be a small scale abstraction of the real world and they are meant to capture the essence of the Hyman P. Minsky theory. We provide also explanations about the identification and the micro calibration of behavioral rules parameters. In other words we detail the work done on point 1 of the list given in the introduction.

2.1 Entrepreneurs Under the Magnifying Glass: Observing Behaviors in the Experimental Environment

According to the experimental method [21], starting from a theoretical model[6] we designed an economic experiment and then develop a software, namely a Graphical User Interface, experimental subjects have to interact with. The software is also used to gather data from experimental sessions.

The experiments took place in September 2011 at the University of Chieti-Pescara (Italy) with the main goal of gaining insights on how experimental subjects playing the role of entrepreneurs manage production and the balance sheet structure in a dynamic and uncertain environment.

Twenty students[7] of the Faculty of Economics and Management, specialized in Accounting and balance sheet analysis, were selected among those who asked to join the experimental program. They took part to six experimental sessions

[5] For an assessment on that criticism see [13] and [18]

[6] Our theoretical model is deeply rooted in the Hyman P. Minsky Financial Instability Hypothesis. In particular our experimental environment reproduces two specific Minskyan features, i.e. the process of decision making under uncertainty and the theory of the firm which takes into account the balance sheet structure.

[7] Some economists have argued that adopting students as the subject pool may compromise external validity of experiments since they are a narrow, unrepresentative segment of the population. However, it is widely acknowledged that there can be problems in using professionals or policymakers since salience (i.e. the reward given to experimental subjects according to the *Induced Value Theory* [20]) is more difficult to establish with such subjects. Experimental economists usually rely on students since they satisfy the three sufficient methodological conditions, i.e. monotonicity, salience and dominance. Moreover, years of experiments have shown that the results are generally insensitive to the choice of subject pool.

according to the scripts (i.e. instructions) they received. Experimental sessions were labelled as $B1a$, $B1b$, $B1c$, $B2a$, $B2b$, $B2c$, $B2$, $B4$, $C1$, $C2$, $D1$ and $D2$. More details on the experimental design and settings can be found in [15] which we briefly summarize hereafter.

The entrepreneur takes in advance a decision about the production whose inventories cannot be carried to subsequent periods. At the beginning of each time step, the number of items ($K_{j,t}$) to be produced is established by guessing the level of demand which will be received. The microeconomic structure makes that the entrepreneur gets the maximum profit when production is equal to demand. Profit is a decreasing function of the guessing error (both excess and shortage) and it becomes negative (the firm suffers a loss) if the deviation is large.[8] If the loss is higher than equity firm must activate a bailout procedure which has a cost. The economic result (profit or loss) and the balance sheet structure of the firm affect each other. We consider a simple balance sheet structure: physical capital ($K_{j,t}$) is the only asset and it can be financed by debt ($B_{j,t}$) or by equity base ($A_{j,t}$). The level of equity base is important and its movement has two opposite effects: on one hand, a high level of equity compensates losses when they are suffered and can avoid bailout procedures; on the other hand, a high level of equity reduces the return on investments (*roi*) because equity substitutes debt which, according to the pecking order theory, is a cheaper source of financing.

The experimental subjects were asked to deal with the just described economic environment by taking two decisions: one about the amount of production and another about the level of equity ratio. Their goal was to maximize a score given by the average *roi* achieved during a sequence of choices minus a penalty for each bailout activated.

2.2 Data Mining

We analyze experimental data in search for models which approximate the experimental subjects' decisions. The effort is to identify the level of a variable (or those of a set of variables) which triggers the change in production and in the liabilities structure. In this paper, we focus on the two best performing experimental subjects which in the experiments were identified by number 11 and 13.[9] Figure 1 shows that they have different equity profiles.

In looking for their behavioral models, we firstly check for rules which are economically grounded. As an example we can imagine the more sophisticated agents estimate the degree of risk by evaluating the volatility of past levels of the return on equity, or we can imagine they use an s-S rule on the level of equity ratio. The goodness of behavioral rules and parameters is evaluated by using the *differential evolution algorithm* [23]. We implement a feedback loop between

[8] By tuning the model parameters we have analyzed how subjects' behaviours concerning production and the balance sheet structure changes under different intensities of the environmental uncertainty.

[9] The experimental data show that some subjects have a "flair" for being entrepreneurs. The two subjects we report on in the paper are the best among them.

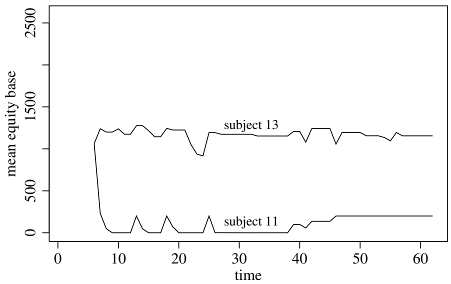


Fig. 1. Series obtained by averaging in each period the equity base chosen in experiments B1b, B1c, B2a, B2b and B2c by experimental subjects 11 and 13

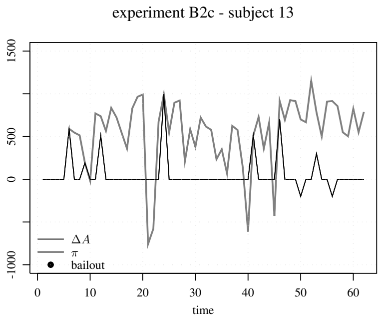


Fig. 2. ΔA and π of experimental subject 13 in experiment B2c

model fitting and model design and we consequently select the models having the best fitting of experimental data. We define this phase the *microeconomic calibration* of subjects' behavior.

In the following sections we will report on the financial choices of experimental agents (the production choice was analyzed in a previous paper. For more details see [17]).

2.3 Microeconomic Calibration of Financial Behavior

In our experiments subjects were asked to manage a simple balance sheet having two liabilities (debt, B, and equity, A). Once the economic result (profit or loss) is computed, each experimental subject was allowed to move the liability structure towards the desired level. The attention is pointed to the movement of the equity base (ΔA_t). The equity adjustment is affected by the economic result: the upward adjustment is possible only when a profit is realized and its amount is bounded by the level of profit; equity base reductions occur when a loss is suffered. In addition, the equity base can always be withdrawn by subjects. Before taking any decision about the movement of the equity base, each subject has to perform the following two-step evaluation:

– setting a target level of equity base ($\hat{A}$);
– deciding whether to move towards the target level.

Concerning the first point, we made an effort to identify rules and parameters governing the choice for the two subjects. The reason for considering the second step is given by possible delays in moving towards the target level. Delays can be due either to the presence of adjustment costs - which make small adjustments not convenient - or just to the waiting behavior for a consolidation of a new detected tendency. In order to account for this eventuality we let the agent start the downward adjustment when

$$A_{j,t} > \hat{A}_{j,t}(1 + d_j^-)$$

and the upward adjustment when

$$A_{j,t} < \hat{A}_{j,t}(1 - d_j^+)$$

where d_j^- and d_j^+ are parameters to be estimated.

Once the adjustment starts, the equity base is moved as fast as possible towards the target level.

In the next subsections we describe the identification of functional forms and parameters calibration of the financial behavior of the selected experimental subjects.

Experimental Subject 13

Functional Forms For Behavioral Rules. We describe in detail the search for the variable(s) which triggers experimental subject financial behavior.

Figure 2 shows the time series of the economic result (π_t) and that of ΔA_t obtained from the choices of experimental subject 13 in experiment B2c. By observing the figure one can see that financial resources are retained (the black line has a peak) after a loss is suffered (the gray line enters the negative portion of the plane). We detected this behavior also in other experimental sessions.

After observing this regularity we conclude that suffering a loss increases this subject's target level of equity and that the following rule:

$$\hat{A}_t = \hat{A}_{t-1}(1 + g_{\hat{A}}^+) \qquad \text{if} \qquad \pi_t < \pi^-,$$

grasps her/his behavior concerning financial resources retention. $g_{\hat{A}}^+$ is a parameter to be estimated.

To complete the investigation for this experimental subject's financial behavior, we have to find out his/her financial resource withdrawal rule. The simultaneous analysis of the ΔA_t, the return on investments (ρ) and the equity base (A_t) time series suggest a "candidate" model for this decision. The three mentioned time series are reported in figure 3 for experiments $B1b$ (left chart) and $B3$ (right chart). There one can observe how this subject leads the equity base towards a target level ($\tilde{A}$) identified by the gray horizontal dashed line. In experiment $B1b$ this process unwinds over the period 14-26 while in experiment

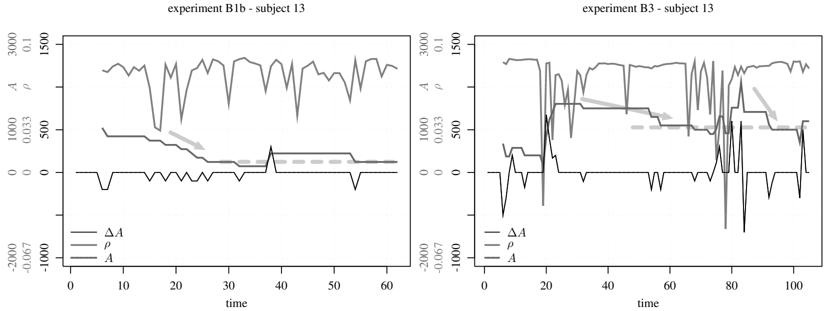


Fig. 3. ΔA, ρ and A of experimental subject 13 in experiment B1b and B3

$B3$ the equity base tends to approach a benchmark level in the time spans 30-70 and 85-100. In those periods the ρ is stable around a high level and occasional troughs (such as the one present in time span 30-70) does not affect the behavior. From these observations we induce that ρ is the triggering variable for the equity base reduction. However, to account for the fact that the subject does not respond to single events we use a moving average (ρ_t^m) as the variable which is responsible for financial resources withdrawals (m is the number of terms the average takes into account). Then, we determine the target level as

$$\hat{A}_t = \hat{A}_{t-1} - g_{\hat{A}}^{-}(\hat{A}_{t-1} - \tilde{A}) \qquad \text{if} \qquad \rho_t^m > \rho^+$$

where $g_{\hat{A}}^{-}$, $\tilde{A}$ and ρ^+ are parameters to be estimated.

Microeconomic Calibration. In order to have a complete identification of our behavioral rules we need to estimate the parameters m, $g_{\hat{A}}^{+}$, ρ^+, d^+, $g_{\hat{A}}^{-}$, ρ^-, d^- and $\tilde{A}$. We also let the algorithm to identify $\hat{A}_0$ which is the initial level of the target equity base we use to take into account the initial transient behavior.

We set the parameters by running the differential evolution algorithm [23]. After studying the results of a Monte Carlo exercise we set the parameters as follows:
$g_{\hat{A}}^{+} = 0.1$, $\rho^+ = 0.066$, $g_{\hat{A}}^{-} = 0.25$, $\rho^- = 0$, $m = 10$, $d^+ = 0.3$, $d^- = 0.3$ and $\tilde{A} = 500$.

The fit of the model can be seen in the left chart of figure 5.

Experimental Subject 11

Functional Forms For Behavioral Rules. This subject shows particular attention to keep the equity base at a low level. In fact s/he sets the equity base to zero unless an "alarm bell" is perceived. This behavior brings him/her to activate the bailout procedure in experiments B1b and B2b. The equity base is set at a positive value, but brought back to zero after these bailouts. This behavior shows that the bailout is not a triggering event for equity base increases.

A close observation of the dynamics of the profit in the periods preceding the first equity base increase in both experiments $B2c$ and $B3$ shed light on the triggering event. By looking at figure 4 one can see how the two patterns of profit are basically the same and they present a number of consecutive reductions of profit. However, in the different sessions there are episodes in which the equity base is increased while the event identified above does not occur (the second episode of the equity base increase in both experiments $B2c$ and $B3$). This reveals the existence of an additional trigger which is found in the equity ratio time series (a_t): both episodes are preceded by a fall of the equity ratio.

Another episode of financial position management is a resource withdrawal in the experimental session $B3$. This can be justified by the profit and ρ levels which are high for a long period.[10] By looking at data one can note how the resource withdrawal causes a trough in the equity ratio which falls below the "safety

[10] The increase of volatility and the deep trough before the event do not affect the subject's behavior.

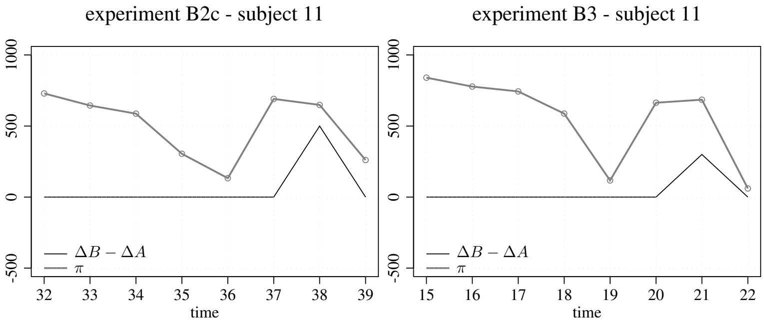


Fig. 4. Zoom on the periods before an increase of financial resources

level". The agent consequently adjusts this figure after a couple of periods by retaining financial resources.

To sum up, there are two triggering events for financial resource retention: a persistent decreasing pattern of profit and a downward threshold of equity ratio. The withdrawal of financial resources is triggered by a long period of high ρ.

To put this subject's rule more formally let us introduce the variable

$$\xi_{12,t}^{m_\xi} := \sum_{i=1}^{m_\xi} \text{sign}(\pi_{12,t-i} - \pi_{12,t-i-1}),$$

so that we can express this subject's behavioral rules as follows:

$$\hat{A}_{12,t} = \hat{A}_{12,t-1} + \Delta\hat{A}^+ \begin{cases} \text{if} \ \ \xi_{12,t}^{m_\xi} = -m_\xi \\ \text{or} \ \ a_t < a^- \ \ \text{and} \ \ a_{t-1} > a_t \end{cases}$$
$$\hat{A}_{12,t} = \hat{A}_{12,t-1} - \Delta\hat{A}^- \ \text{if} \ \ \rho_{12,t}^{m_\rho} > \rho^+$$

Microeconomic Calibration. To make these rules working, we need to estimate the parameters m_ρ, m_ξ, $\Delta\hat{A}^+$, ρ^+, $\Delta\hat{A}^-$, a^-, d^+ and d^-. For this subject

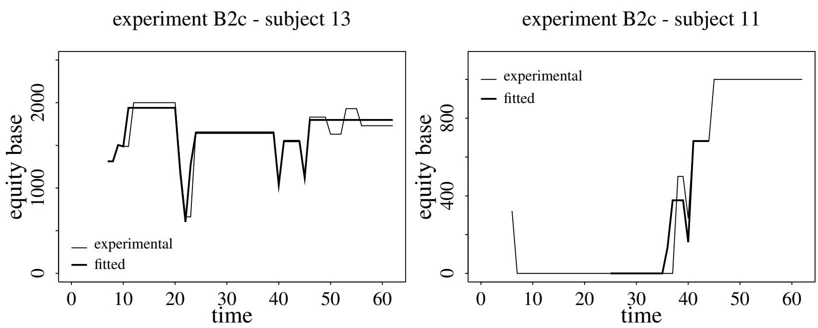


Fig. 5. The equity base behavior of the subjects we are studying (thin line) and the fitted values produced by the model (thick line)

we have informative movements of the financial position in the time span 25-44 of experiment B2c and in the whole experiment B3.

After studying the results we set the parameters as follows:
$m_\rho = 20$, $m_\xi = 5$, $\Delta\hat{A}^+ = 200$, $\rho^+ = 0.086$, $\Delta\hat{A}^- = 50$, $a^- = 0.03$, $d^+ = 0.3$ and $d^- = 0.3$.

The fit of the model can be seen in the right chart of figure 5.

3 The Multi-avatar Macroeconomy

In the previous section we have identified rules and parameters which regulate the behavior of the two categories of entrepreneurs we have studied, focusing in this way on point 1 of the list given in the introduction. This section focuses on the other items. By using the object oriented programming[11] we build a class for each type of entrepreneur and we populate the economy with a large number of instances of these classes.[12]

At each simulation step, each artificial agent is involved in the following:

– s/he determines the level of production by using a one-step-ahead forecasting module (see [17] for details on this module);
– the level of demand ($y_{j,t}$) gets known;
– the economic result is computed (see [16] for details);
– the financial position is managed by using the rules identified in the previous section.

To close the model we have to establish how the level of demand ($y_{j,t}$) is set for each entrepreneur.

The Level of Demand for Each Artificial Agent

In our context, the "natural" way to deal with this issue would be the creation of consumer avatars. Since this phase of our research is still in progress we follow a method implemented in [14] which is based on the following reasoning. First, demand depends on households' income. Second, income is equal to the value of production.

Let us start from the previous period aggregate production Y_{t-1}. When a firm asset becomes lower than debt due to losses, agents are allowed to activate a bailout procedure which generates an income loss (because for example a number of workers are dismissed). The aggregate income loss (Y_t^-) is endogenously determined as a percentage (θ) of assets of firms which activate the bailout.

On the other hand, firms profits increase the income of the economy. The aggregate income gain (Y_t^+) is given by a percentage (η) of alive firms profit.

The aggregate demand has the usual dynamics:

$$Y_t = Y_{t-1} - Y_{t-1}^- + Y_{t-1}^+.$$

[11] We are developing in Java taking advantage of the RepastJ 3.1 classes.

[12] Presently we have developed two classes: one for subject 13 and the other for subject 11. An ongoing work consists in "enriching" the artificial economy with additional experimental subjects' avatars.

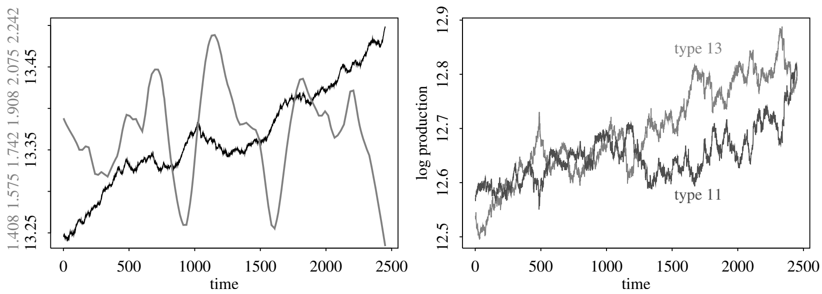


Fig. 6. Left chart: aggregate production (black line) and nonparametric regression of the number of bailouts (gray line). Right chart: market shares of the two categories of entrepreneurs.

The final step is to obtain the firm level demand $(y_{j,t})$ from the aggregate level (Y_t) . Having no product differentiation, each unit of the aggregate demand is randomly allocated to one of the existing firms.

Setting Macroeconomic Parameters. Monte Carlo runs of the macroeconomic agent based model allow to select the settings which give real like patterns for macroeconomic variables. For a model having 200 firms (100 firms for each type of financial behavior we defined in subsection 2.3) the Monte Carlo exercise brings us to the following settings: $\eta = 0.017$; $\theta = 0.1$; assets at the beginning (K_0) and after bailout is set to 3000 and the equity ratio at the beginning (a_0) and after bailout is set to 0.05.

Simulation Results. The black line in the left chart of figure 6 shows the logarithm of the aggregate production of our multi-avatar economy. The time series has a "real like" pattern with expansions and recessions showing different length and intensity. An interesting exercise could be done by comparing the aggregate output with the gray line which gives the unfolding of the number of firms bankruptcy (this line is obtained by applying nonparametric smoothing techniques). It can be observed, in particular by looking at the two series around period 1000, that the number of bailouts increases during the expansion period. When the business cycle reverts, firms financial position keeps worsening for a while before reverting its slope. This pattern calls to mind Minky's "Financial Instability Hypothesis" according to which agents adopt more "dangerous" financial behavior during expansions and this makes capitalistic economies intrinsically unstable.

An important work on simulated data analysis and their comparisons with real world data can be performed as suggested by the Agent-Based literature. Starting from the data recorded at the individual level from artificial agents, we can keep track of the dynamics of the economy at different levels of aggregation (see [7] among others). As a first example we trace the dynamics of the market shares of the two types of entrepreneurs populating the economy (see right chart of figure 6) which could help understanding which agent type contributes the most to the different cyclical phases. A second example is related to the study of the dynamics of the distributions of individual variables (firm size, equity base, equity ratio, etc.) for each type of agent or for the whole economy. The kind of analysis on simulations results just mentioned in the two examples above and the possibility to use the model for policy purposes are beyond the scope of this paper. However, we maintain that they will be useful for policy analyses and represent a promising point of departure for future research.

4 Conclusions

One of the most likely difficulties in the building of an Agent-based Computational Economics model is the setting of the behavioral rules for artificial agents. The *deductive method* based on the axiomatic approach, although appropriate in a “perfect” economic context (i.e. General Equilibrium models), can be flawed in replicating real life agents' behavior. The *inductive approach* proposed and described in this paper, although not free from weakness, could provide a valid alternative to the empirical grounding of macroeconomic computational models.

The mentioned weakness is mainly related to the identification of recurrent behaviors by using experiments. In order to check for the internal and external validity of the experimental design and results used to calibrate artificial agents, we are running experimental sessions with a different subject pool, i.e. entrepreneurs. Some early results show that there are regularities in the financial behaviors observed in the laboratory. Although analyses and investigation are still on the way, those early results could be interpreted as an evidence for a good generalization of laboratory regularities to the larger economy outside the laboratory. In this sense our work could represent an answer to the claim that existing ABMs still lack generally accepted microfoundations because of the “excess freedom” Agent Based researchers face.

In our research we strive to shed new light on this issue by providing a new methodology that helps significantly reducing those difficulties by relying on a combination of the experimental method and Agent-based modeling tools.

The model presented herein is simple and can be extended in several directions. Moreover, we believe the proposed approach will open promising perspectives for macroeconomic theory and could yield insights that may be elusive in more conventional frameworks of analysis and therefore could “produce” more reliable models for policy purposes.

References

1. Akerlof, G.A.: Behavioral macroeconomics and macroeconomic behavior. American Economic Review 92(3), 411–433 (2002), `http://ideas.repec.org/a/aea/aecrev/v92y2002i3p411-433.html`
2. Buchanan, M.: Meltdown modelling. Nature 460, 680–681 (2009)
3. Camerer, C., Loewenstein, G., Rabin, M.: Advance in Behavioral Economics. Princeton University Press (2004)

4. Chari, V.V., Kehoe, P.J.: The heterogeneous state of modern macroeconomics: A reply to Solow. Staff Report 399, Federal Reserve Bank of Minneapolis (2007), http://ideas.repec.org/p/fip/fedmsr/399.html
5. Colander, D., Howitt, P., Kirman, A., Leijonhufvud, A., Mehrling, P.: Beyond DSGE Models: Toward an Empirically Based Macroeconomics. American Economic Review 98(2), 236–240 (2008), http://www.aeaweb.org/articles.php?doi=10.1257/aer.98.2.236
6. Dawid, H., Neugart, M.: Agent-based models for economic policy design. Eastern Economic Journal 37(1), 44–50 (2011), http://ideas.repec.org/a/pal/easeco/v37y2011i1p44-50.html
7. Delli Gatti, D., Gaffeo, E., Gallegati, M., Giulioni, G., Kirman, A., Palestrini, A., Russo, A.: Complex dynamics and empirical evidence. Information Sciences 177, 1204–1221 (2005)
8. Dosi, G., Fagiolo, G., Roventini, A.: Schumpeter meeting Keynes: A policy-friendly model of endogenous growth and business cycles. Journal of Economic Dynamics and Control 34(9), 1748–1767 (2010), http://ideas.repec.org/a/eee/dyncon/v34y2010i9p1748-1767.html
9. Duffy, J.: Agent-based models and human subject experiments. In: Tesfatsion, L., Judd, K.L. (eds.) Handbook of Computational Economics, Handbook of Computational Economics, vol. 2, ch. 19, pp. 949–1011. Elsevier (2006), http://ideas.repec.org/h/eee/hecchp/2-19.html
10. Duffy, J.: Macroeconomics: A survey of laboratory research. Working Papers 334, University of Pittsburgh, Department of Economics (February 2008), http://ideas.repec.org/p/pit/wpaper/334.html
11. Fagiolo, G., Birchenhall, C., Windrum, P.: Empirical Validation in Agent-based Models: Introduction to the Special Issue. Computational Economics 30(3), 189–194 (2007), http://www.springerlink.com/index/10.1007/s10614-007-9109-z
12. Farmer, J.D., Foley, D.: The economy needs agent-based modelling. Nature 460, 685–686 (2009)
13. Foley, D.K.: Hyman Minsky and the dilemmas of contemporary economic method (1998)
14. Giulioni, G.: The product innovation process and GDP dynamics. Journal of Evolutionary Economics 21, 595–618 (2011)
15. Giulioni, G., Bucciarelli, E., Silvestri, M.: An economic experiment to investigate firms financial decisions. Mimeo (2012), http://www.dmqte.unich.it/users/giulioni/experiment_sept_2011.pdf
16. Giulioni, G., Bucciarelli, E., Silvestri, M.: Exploring firms financial decisions by human and artificial agents. In: Filipe, J., Fred, A. (eds.) Proceedings of 4th International Conference on Agents and Artificial Intelligence, ICAART, pp. 184–189 (2012)
17. Giulioni, G., Bucciarelli, E., Silvestri, M.: Modeling firms financial behavior in an evolving economy by using artificial intelligence. In: Pérez, J.B., Sánchez, M.A., Mathieu, P., Rodríguez, J.M.C., Adam, E., Ortega, A., Moreno, M.N., Navarro, E., Hirsch, B., Lopes-Cardoso, H., Julián, V. (eds.) Highlights on PAAMS. AISC, vol. 156, pp. 333–340. Springer, Heidelberg (2012)
18. Goodhart, C.: Minsky I find enormously attractive but his issues are very difficult to model in any rigorous way. European Journal of Economics and Economic Policies: intervention 6(1), 5–11 (2009)
19. Minsky, H.P.: Stabilizing an Unstable Economy. Yale University Press, New Haven (1986)

20. Smith, V.L.: Experimental economics: Induced value theory. American Economic Review 66(2), 274–279 (1976), http://ideas.repec.org/a/aea/aecrev/v66y1976i2p274-79.html
21. Smith, V.L.: Method in experiment: Rhetoric and reality. Experimental Economics 5, 91–110 (2002), http://dx.doi.org/10.1023/A:1020330820698, 10.1023/A:1020330820698
22. Solow, R.: The state of macroeconomics. The Journal of Economic Perspectives 22(1), 243–249 (2008), http://www.jstor.org/stable/10.2307/27648233
23. Storm, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4), 341–359 (1997)
24. Tesfatsion, L., Judd, K.: Handbook of Computational Economics. Agent-Based Computational Economics, vol. 2. Elsevier (2006)

Dynamic Filtering of Useless Data in an Adaptive Multi-Agent System: Evaluation in the Ambient Domain

Valérian Guivarch, Valérie Camps, André Péninou, and Simon Stuker

Institut de Recherche en Informatique de Toulouse
{Valerian.Guivarch,Valerie.Camps,Andre.Peninou,Simon.Stuker}@irit.fr
http://www.irit.fr

Abstract. *Amadeus* is an Adaptive Multi-Agent System whose goal is to observe and to learn users' behaviour in an ambient system in order to perform their recurrent actions on their behalf. Considering the large number of devices (data sources) that generally compose ambient systems, performing an efficient learning in such a domain requires filtering useless data. This paper focuses on an extended version of *Amadeus* taking account this requirement and proposes a solution based on cooperative interactions between the different agents composing *Amadeus*. An evaluation of the performances of our system as well as the encouraging obtained results are then shown.

Keywords: Adaptation, Learning, Distributed-problem solving, Data filtering, Pervasive agents, Ambient intelligence.

1 Introduction

The performances of learning algorithms is generally degraded by the presence of useless data among the ones used to learn, a piece of data being considered to be useless if there is no link between its value and the objective to learn. One way to improve this fact is to select useful data. An ambient system is composed of many heterogeneous devices, often mobile, physically distributed and interacting in a dynamic way. So, it is a typical example where applying a learning is a very complex task, because it consists of a great number of devices that are able to generate data. Furthermore, devices may appear and disappear dynamically. Thus, in this case, the filtering of data coming from these devices cannot be defined *a priori*, that is before the learning process. Learning has to be done at runtime, without restarting from scratch when new data appear. The filtering of useless data has also to be done at runtime.

This is particularly the case for the multi-agent system *Amadeus* [4] whose goal is, in an ambient system, to observe the users' actions in order to learn those that are recurrent and to learn then how to perform them on behalf of the users. This learning is performed in a decentralized way, an instance of *Amadeus* being responsible of each device of the ambient system. However, the large number of

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 110–121, 2013.

devices requires the filtering of useless data at runtime for the users' behaviour learning.

In this paper, we present an extended version of *Amadeus*. Section 2 briefly presents the general functioning of *Amadeus*. Our contribution enabling the "on-line" filtering of useless data is presented in section 3 and evaluated in section 4. Section 5 is devoted to related works. Section 6 concludes this paper and explains the on-going work.

2 Our MAS Proposition for Ambient System: *Amadeus*

Our contribution aims at proposing a solution to tackle the problem of adaptation in ambient systems. We propose to make an ambient system able to provide a relevant behaviour, based on the perceived user's actions, in order to assist him by realizing his actions on his behalf. We have already proposed *Amadeus* [4], an Adaptive Multi-Agent System that is able to learn the user's contexts while this user is performing recurrent actions in order to act on his behalf in similar situations.

An instance of the *Amadeus* multi-agent system is associated to each device of an ambient system. Figure 1 is a representation of an instance of *Amadeus*. We can observe the four types of agents existing in the system: *Data* agents, *User* agents, *Context* agents and *Controller* agents.

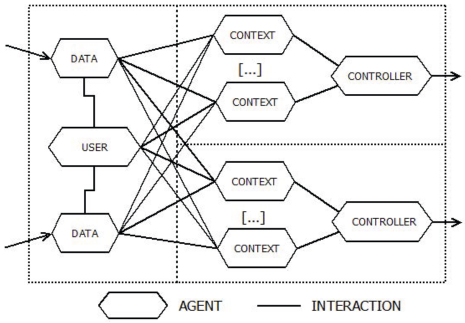


Fig. 1. Representation of an instance of *Amadeus*, one instance is associated to each device of an ambient system

A *Controller* agent is associated to each effector of a device. Its goal is to decide at anytime what is the best action to perform on the effector on behalf of the user. This decision is made thanks to a set of *Context* agents.

A *Context* agent is created each time a user performs an action in his environment (for example to turn on the light). This agent associates this action with a description of the situation in which the user performs this action. This situation is composed of the set of the perceived data values when the action is performed (example: Light = 0 ; PresenceSensor= 1 ; LuminositySensor = 22).

The action is represented by the value given to the effector (for example, 1 to turn on and 0 to turn off the light).

The *Context* agent adds to this description a forecast on the effect of this action on the user's satisfaction level (does the completion of this action for this situation lead to increase, to maintain or to decrease the user's level satisfaction?). This forecast is obtained by comparing the user's satisfaction level before and after the achievement of the action. This satisfaction is represented by a value between 0 (the user is not satisfied at all) and 1 (he is very satisfied), and is estimated by the *User* agent. The *User* agent is responsible for the user's preferences. Thanks to a representation of the user's preferences (currently with a XML file) the *User* agent can evaluate, for any situation, if the user is satisfied with regard to the state of the device effector. One of our work perspectives is to make the *User* agent able to dynamically and autonomously learn these preferences, but this is out of the scope of this paper.

Every *Context* agent perceives a set of data coming from local sensors situated on either the same device or on distant sensors situated on other instances of *Amadeus* (namely on devices). The *Data* agents represent these data. In the *Context* agent, each of these data possesses a validity status that depends on its current value with regard to the situation described (by the *Context* agent). A data is considered as valid if it is included in a values range. This range represents the values interval that a piece of data may have in order to describe a situation. Thus, the *Context* agent tries to establish the borders of valuable ranges for every perceived data that enable it to describe the correct situation for which its action proposition is appropriate (namely it will have the expected effects). To do this, the *Context* agent possesses, for each perceived data, an Adaptive Range Tracker (ART) that is a data structure enabling to describe a valuable interval (called "interval of validity") where min and max borders can evolve. The value of each border is estimated with an Adaptive Value Tracker (AVT) [11], which is a tool devoted to the tuning of evolving parameters. A *Context* agent considers a data as valid if its current value belongs to its associated ART. A *Context* agent has also its own validity status. Its status is valid if all perceived data are valid (it is invalid otherwise). In this case, a *Context* agent sends its action proposition and its forecast to the *Controller* agent. The *Controller* agent can then decide which action, among those proposed by all valid *Context* agents, is the most appropriate to satisfy the user.

A first evaluation of this *Amadeus* version applied to a simple example gave us encouraging results [4] [5]. Nevertheless, we observed a strong weakness regarding the required learning time when the number of perceived data increases. In particular, the addition of "useless" data that change independently of the user's actions on an effector, implies a strong slowing down of the learning time. Such data are perceived by *Context* agents, and so are included in the situation description, but they do not affect the user' behaviour. For example, *Context* agents can perceive humidity level of the rooms but it is not necessary to consider this fact to decide to turn on or to turn off the lights. Indeed, if a situation having previously led to the realization of an action by the user appears again, with useless piece of data having a different status, the system considers wrongly that the situation is different. For example, if the system has learnt to switch on the light when the user goes into the room but with a specific level of humidity, when the user enters again into the room, a change of humidity level perceived by a humidity sensor leads the system to consider itself in a new situation, and thus to not act.

To overcome this problem, it is necessary to determine, for each effector, which data are useless for learning the behaviour to give to this effector, in order to only consider the useful data. Describing explicitly what are the useful and the useless data for each effector seems to be a limited solution, because of the strong dynamic of an ambient system. This is why we propose to make each instance of *Amadeus* able to autonomously determine, without any *a priori* information and at runtime, what are the data useless to learn the user's behaviour.

3 Data Filtering

The objective of our study is to locally detect what are the useless data for a device. A piece of data is considered as useless if its value is independent of the user's decision to act on this device. This detection is based on a learning performed at runtime, without any *a priori* information.

3.1 Proposed Approach

Our proposition to detect useless data is based on the cooperation between agents. On the one hand, the *Context* and *Controller* agents are responsible for the user's behaviour learning towards the state of an effector. On the other hand, the *Data* agents are responsible for providing useful data to *Context* and *Controller* agents so that they make their learning. Thus, *Data* agents and *Context* agents have to interact for determining which data are useful to characterise the situation in which the user acts on the device, and which data are useless.

A *Context* agent is created each time the user performs an action on an effector. This agent associates the user's action with a description of the situation in which the user had performed it (see section 2). This situation is composed of the set of the perceived data values when the action was performed. The assumption is that all perceived data contribute to characterize the situation. Thus, later, when the *Context* agent becomes valid, it can be sure of its decision. But what about when it is not valid? Is it invalid because all data contributes really to describe the situation (it is right to be invalid) or because one useless data possesses a current value making the agent invalid (it is wrong to be invalid)? The *Context* agent cannot solve this ambiguity by itself. Nevertheless, a more interesting point of view is to think about useful and useless data when at once i) the *Context* agent is invalid and ii) another one is valid. In such cases, the *Context* agent can question itself about the necessity to be invalid in the current situation. We use these cases in our approach described hereafter.

When a *Context* agent is selected and its action is made by the *Controller* agent (or when the user makes an action himself), every invalid *Context* agent observes its internal state, and evaluates if it was right to do not send its action proposition. More details on this evaluation are given in section 3.2. When a *Context* agent establishes it was right to be invalid at a given time, it was thanks to the invalid perceived data. So, among these currently invalid data, it knows that there is at least one useful piece of data. However, without more information, the *Context* agent cannot determine if a data is invalid because it is really useful, or if it is just a coincidence that this data is invalid in the current situation. So, it sends a message called a "*usefulness signal*" to all the data that are invalid in order to inform them that they are potentially useful.

In order to evaluate the usefulness of its data, each *Data* agent perceives the *usefulness signals* sent by all the *Context* agents. The goal of a *Data* agent is to process these signals in order to establish if the reception of these usefulness signals is just the result of coincidences (the associated data was invalid at the good moment, but even if it had not been, a data really useful would have been invalid) or if it was invalid at the good moment because it is a useful data. This process is described in section 3.3.

3.2 Usefulness Signal Generation

Let us consider a *Context* agent C. Another *Context* agent S has just been selected, whereas C was invalid, because among the set of perceived data D, there is a subset of invalid data D_I for C. So, agent C observes the selected agent S to know if it proposed the same action or not, and if this proposition was or not associated with the same forecast on the effect of this action on the user's satisfaction level. Two particular cases can be highlighted.

The first one occurs when C and S propose the same action (for example, turn on the light) but with different forecasts. It is obvious that the same action cannot have two different effects in the same situation. Then, the *Context* agent C was right to be invalid when S was valid. So, at least one of the invalid data of C is useful. Let us give an example for this case. S proposes to turn on the light when someone is in the room with a low luminosity level and a high humidity level. On the contrary, C proposes to turn on the light when someone is in the room with a high luminosity level and a low humidity level. S forecasts an increase of the user's satisfaction if it is selected, whereas C forecasts a decrease of the user's satisfaction if it is selected (to turn on the light if the luminosity is high is not satisfying for the user). When S is selected, the situation is: someone is in the room, low luminosity level and high humidity level. C observes that it proposes the same action as S, but with a different forecast. So it can consider that in the set of invalid data composed by the luminosity and the humidity levels, there is at least one useful data for it (in this case, the luminosity data).

The second case occurs when C and S propose two different actions with the same forecast. So, if S proposes an action that increases the user's satisfaction level, C cannot be valid in the same situation if it proposes a different action that also increases the user's satisfaction level. This is why C can consider that, in the set of its invalid data, at least one is useful. Let us give an example for this case. S proposes to increase the user's satisfaction level by turning on the light when the user is in the room with a low luminosity level and a high humidity level. On the contrary, C proposes to increase the user's satisfaction level by turning off the light when the user is in the room with a high luminosity level and a low humidity level. When S is selected, the situation is: the user is in the room, low luminosity level and high humidity level. C evaluates that it cannot be valid at the same time, so it considers that, in the set of its invalid data composed by the luminosity level and the humidity level, there is at least one useful data.

For each of these situations, C sends a *usefulness signal* to each of its invalid data in order to warn them that they are "maybe" useful.

To conclude, *Context* agents are able to detect and gather situations or information about the usefulness of perceived data. However, these information concern the usefulness of data, whereas we are interested in their uselessness. So, in the next section, we describe how the *Data* agents process the *usefulness signal* to detect if they are useful data or not.

3.3 Usefulness Signal Processing

First of all, let us underline that *Context* agents and *Controller* agents are always bound to a single effector of a device (a complex device may have different effectors). *Data* agents receive *usefulness signals* that are implicitly bounded to an effector. Thus, *Data* agent processing of these signals must be performed separately for each effector.

We consider two data F and L, where F is useful and L is useless with respect to an effector E. Also, we consider two sets of *Context* agents $SC1$ and $SC2$, where the agents of $SC1$ propose to switch the effector to some state e_1, and the agents of $SC2$ propose to switch the effector to some other state e_2. Each time a *Data* agents (F or L) receives a *usefulness signal*, it observes its current value and the state of the effector E. With these values, it computes a density function of the values taken at the reception of *usefulness signals* regarding the effector state proposed by the *Context* agent that sent the signal. Let S_E denotes the set of possible states of the effector E. So, for each state e in S_E, and each *Data* agent D, $d_D(e)$ denotes the density function of agent D with respect to effector's state e when *usefulness signals* are received.

The distinction between the useful data F and the useless data L can be observed through the density functions $d_F(e)$ and $d_L(e)$. Because *Data* agent F is useful, it is correlated with the actions applied on the effector, hence $d_F(e_1) \neq d_F(e_2)$. Conversely, *Data* agent L is useless and has no influence on the effector's actions. We can observe this fact through the similarity between $d_L(e_1)$ and $d_L(e_2)$.

On the basis of these remarks, we can evaluate the usefulness of a *Data* agent by comparing the density functions corresponding to the different states of the effector. More precisely, the distance between two density functions $d_D(e_1)$ and $d_D(e_2)$ is measured through the *Chi-square distance* [3] relative to the general data frequency:

$$\delta\left(d_D(e_1), d_D(e_2)\right) = \sum_{Value\ i} \frac{(d_i - d'_i)^2}{d'_i}$$

where d_i (respectively d'_i) denotes the frequency of value i for the *Data* agent D (whenever D is considered useful by *Context* agents) with respect to the effectors state e_1 (respectively e_2). The usefulness of a *Data* agent D, $U_D(E)$, can then be expressed as the maximum distance between any pair of its density functions:

$$U_D(E) = \max_{x,y \in S_E} \left(\delta\left(d_D(x), d_D(y)\right) \right)$$

The use of the *chi-square distance* in order to compare two density functions allows obtaining a value that has a statistical significance. As a matter of fact, under the assumption that the data is useful, $\delta\left(d_D(e_1), d_D(e_2)\right)$ follows some *chi-square* law [1]. So, we can grant a statistical credibility to the evaluation of the usefulness data.

When a *Data* agent receives a *usefulness signal*, it computes its density functions with respect to the effector states. So, the evaluation of its usefulness based on these density functions becomes more and more precise. Finally, when the usefulness level of a *Data* agent gets below a certain fixed threshold empirically calculated, the *Data* agent considers that it is useless for the effector.

In this case, the *Data* agent informs the set of *Context* agents associated to this effector of its uselessness, and then its stops to send them update about its value. Then the *Context* agents delete the ART associated with this data, and forget this data to estimate their validity state.

Our filtering is based on the signal sent by the *Context* agents, each signal concerning a set of *Data* agents. So, contrary to classical methods that filter each data independentely, this method takes care about dependencies between data.

4 Experimentations

The proposed solution was implemented using Speadl/MAY [12], which is a tool allowing to assemble reusable components in order to build architectures supporting the development and execution of multi-agent systems. Our solution was evaluated through a simulator using users' preferences (given in a XML file) to generate users' behaviour in a virtual ambient system.

Our experimentation takes place in a case study composed of an apartment with five rooms (one living room, one bathroom, one kitchen and two bedrooms). Each room possesses a light, a luminosity sensor and a presence sensor. Each *Amadeus* instance associated to a light effector has to learn a good behaviour based on the user's actions on this effector. Each instance perceives not only the data from its room (the luminosity sensor, the presence sensor, and the state of the light), but also the data of the other rooms. Among the fifteen perceived

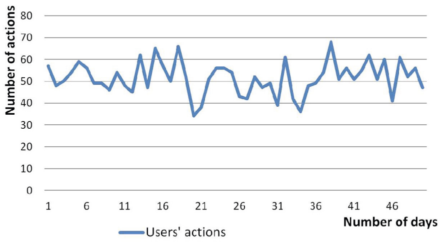


Fig. 2. Number of users' actions per day without *Amadeus*

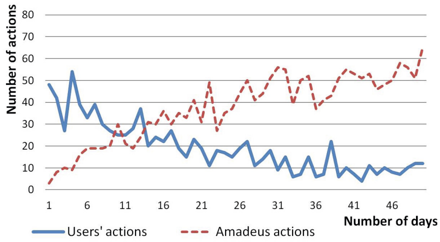


Fig. 3. Number of users' actions and *Amadeus* actions per day without filtering data

data (three by room and five rooms), only three data are useful for each instance, and twelve are useless.

We added three users to this simulation. These users can move between the different rooms; they can also leave (and come back to) the apartment (we consider a sixth room called "outside" without any sensor and effector). The users' behaviour is based on simple rules: when a user enters in a room with a switched off light and an insufficient luminosity, he turns on the light, whereas if the luminosity is very strong with a switched on light, the user turns off the light. When he leaves a room, if he was the last in this room and the light was switched on, he turns off the light. Figure 3 represents the number of users' actions per day during a simulation of fifty days, without the *Amadeus* use. We can observe an average number of 50 actions per day made by the different users.

The first experiment consists in adding an instance of *Amadeus* (first version without useless data filtering) to each device of our case study. Every instance associated with the light device is in charge to learn the correct behaviour to give to this light device according to the users' actions. Figure 3 represents the number of actions respectively made by the users and *Amadeus* on the different devices during a simulation of fifty days. We can observe that, even if *Amadeus* makes many actions on behalf of the users, its performances are very limited and the users have to make a lot of actions even after 50 learning days. This can be explained by the fact that each *Amadeus* instance responsible for each light device has difficulties to learn in which situation every action is realized, because of the too numerous useless perceived data.

The second experiment consists in carrying out the same experiment with *Amadeus* instances able to filter useless data. So, based on the process explained in section 3, the *Data* agents are able to locally and autonomously detect what are the useless perceived data for each device. For each of the five effectors, 12 useless data have to be filtered (hence a total number of 60 useless data for the entire system). After fifty days of simulation, the system has filtered 48 useless data (11 for the first light, 9 for the second, 8 for he third, 9 for the fourth and 11 for the fifth). Figure 4 shows the *Amadeus* capabilities to make actions on behalf of the users with the use of filtering useless data, only 15 days being necessary to decrease users' actions to less than 10 per day.

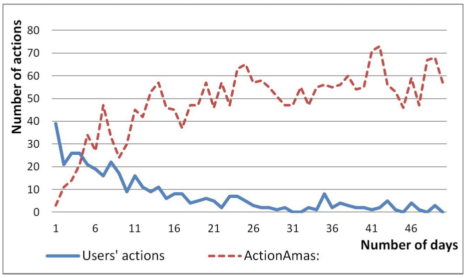


Fig. 4. Number of users' actions and *Amadeus* actions per day with filtering data

This simulation has been made twenty times, in order to evaluate the quality of the filtering of useless data on several simulations. Each simulation gives a quite random behavior to each user for moving in the apartment. Table 1 displays obtained results. The first line shows the total number of filtered data whereas the second line shows the number of useful data wrongly filtered. The third and fourth lines respectively show the percentage of useless data filtered, and the percentage of useful data wrongly filtered. We obtained a final average percentage of useless data filtered equal to 79.7, and an average percentage of useful data wrongly filtered equal to 1.

Table 1. Results of the evaluation of *Amadeus* for twenty simulations

Total number of filtered data	55	42	51	41	48	52	46	56	50	49
Number of useful data wrongly filtered	0	0	0	0	0	0	0	0	0	1
Percentage of useless data filtered	91,7	70	85	68,3	80	86,7	76,7	93,3	83,3	81,7
Percentage of useful data wrongly filtered	0	0	0	0	0	0	0	0	0	6,7

Total number of filtered data	48	38	42	45	49	49	50	49	47	49
Number of useful data wrongly filtered	0	0	0	0	0	1	0	0	1	0
Percentage of useless data filtered	80	63,3	70	75	81,7	81,7	83,3	81,7	78,3	81,7
Percentage of useful data wrongly filtered	0	0	0	0	0	6,7	0	0	6,7	0

The data selection performed by the *Data* agents is not perfect because about 20% of useless data remain unfiltered. However, it is possible to decrease the chosen threshold to decide data uselessness in order to have better results regarding the data filtering, but it would imply an increase of the rate of wrong filtering. Nevertheless, our objective is not necessary to filter all the useless data, but to filter a sufficient number of useless data in order to make *Amadeus* able to learn the users' behaviour. Hence, we consider that it is better to have unfiltered useless data than wrongly filtered useful data.

5 Related Work

A very explicit illustration of the useless data effect on learning algorithms is given by [2]. Figure 5 shows an example where a learning algorithm tries to classify information into two classes. If the learning algorithm observes the A and B points using only the data x_1 (represented by their projection A' and B' on the x_1 axis), it will find correctly the two classes (dotted vertical line). However, if it considers the useless data x_2, it will fail to separate the two classes (diagonally continuous line). So, considering useless data in the learning process makes it necessary to increase the number of examples that must be provided to the learning algorithm in order to overcome this problem.

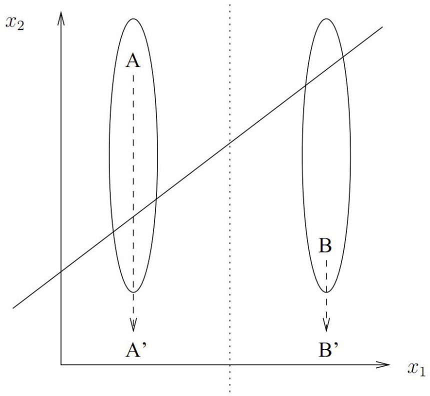


Fig. 5. Illustration of the effect of useless data on a learning algorithm (from [2])

In the literature, many solutions were proposed to solve this very recurrent problem in learning methods [7]. A first solution concerns variable ranking methods [6]. Such methods try to compute a score for each data that represents the usefulness of this data with respect to the target of the learning. This evaluation is performed independently of each data, and data with a low score are considered as useless. However, in complex systems such as ambient systems, the large number of interactions between data makes such methods inappropriate. As a matter of fact, the effect of a piece of data on the target of the learning can be strongly dependant on other data, so an evaluation of the usefulness of this piece of data independently of other data is not appropriate [14]. For example, in the study of the section 4, the usefulness of luminosity sensor is strongly depending of the value of presence sensor, so it is necessary to evaluate the usefulness to this data regarding the values of other data.

Other methods allow the selection of useful data by considering subsets of data rather than independent pieces of data. These methods can be divided into three categories [13]:

1. *Filter* techniques that carry out the selection of useful data independently of the learning algorithm itself; they are unable to interact with this one during learning process [9].
2. *Wrapper* techniques that use the learning algorithm itself in order to evaluate the usefulness of subsets of data. For this, the learning algorithm is applied to different subsets of data, and the performances of the learning are evaluated regarding the data subset to which it is applied. They generally get better results than *Filter* techniques, but they require much more computation time and they present a strong risk of overfitting [8].
3. *Embedded* techniques that are directly integrated in the learning algorithm itself (the variable selection being strongly correlated with the process of the learning algorithm). For example, some algorithms observe the impact of the addition or the removal of data in their learning process in order to evaluate the usefulness of these data [10].

Because of the dynamic and distributed properties of the learning algorithm implemented by *Amadeus*, we considered, as we implemented it, that an embedded technique of data selection was the most appropriate solution.

6 Conclusion

In this paper, we have presented an extended version of the multi-agent system *Amadeus*. This system is devoted to the learning of users' behaviour in ambient systems. *Amadeus* can learn situations and actions performed by the user on effectors of devices in order to perform later these actions on behalf of the user. However, the large number of useless data, because of the high number of devices in ambient systems, makes necessary to improve *Amadeus* learning by adding filtering data capabilities. We have introduced a new ability to the *Data* agents, which purpose is to detect if a piece of data is useless for every effector of a device in the ambient system or not. We also described the data filtering process performed by the *Data* agents, and presented first results about the *Data* agents' filtering performances.

The choice to define a threshold to decide of the uselessness of each data gives encouraging results. For example, *Amadeus* is able to filter a large part of useless data, and it wrongly filters a very low level of useful data. However, in an adaptive multi-agent system, having a static parameter is a weakness in term of real adaptation. That is why we are currently investigating a solution to make the *Data* agents able to filter useless data and to autonomously and dynamically define the proper threshold to use.

References

1. Baillargeon, G.: Introduction à l'inférence statistique: méthodes d'échantillonnage, estimation, tests d'hypothèses, corrélation linéaire, droite de régression et test du khi-deux avec applications diverses. Techniques statistiques. Les Editions SMG (1982)

2. Delalleau, O.: Extraction hiérarchique de caractéristiques pour l'apprentissage à partir de données complexes en haute dimension, Pre-doctoral report, University of Montreal (2008)
3. Greenwood, P.E., Nikulin, M.S.: A Guide to Chi-Squared Testing. Wiley, New York (1996)
4. Guivarch, V., Camps, V., Péninou, A.: Context awareness in ambient systems by an adaptive multi-agent approach. In: Paternò, F., de Ruyter, B., Markopoulos, P., Santoro, C., van Loenen, E., Luyten, K. (eds.) AmI 2012. LNCS, vol. 7683, pp. 129–144. Springer, Heidelberg (2012)
5. Guivarch, V., Francisco De Paz Santana, J., Bajo Pérez, J., Péninou, A., Camps, V.: Learning user's behaviour with an Adaptive Multi-Agent System approach (regular paper). In: Intelligent Systems for Context-based Information Fusion, Cartagena de Indias - Colombia, Springer (2012)
6. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)
7. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.A.: Feature extraction: foundations and applications, vol. 207. Springer (2006)
8. John, G.H., Kohavi, R., Pfleger, K., et al.: Irrelevant features and the subset selection problem. In: Proceedings of the Eleventh International Conference on Machine Learning, San Francisco, vol. 129, pp. 121–129 (1994)
9. Kira, K., Rendell, L.A.: The feature selection problem: Traditional methods and a new algorithm. In: Proceedings of the National Conference on Artificial Intelligence, p. 129. John Wiley & Sons Ltd. (1992)
10. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
11. Lemouzy, S., Camps, V., Glize, P.: Principles and Properties of a MAS Learning Algorithm: a Comparison with Standard Learning Algorithms Applied to Implicit Feedback Assessment (regular paper). In: IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT), Lyon, pp. 228–235. CPS (Conference Publishing Services) (août 2011)
12. Noel, V.: Component-based Software Architectures and Multi-Agent Systems: Mutual and Complementary Contributions for Supporting Software Development. Thése de doctorat, Université de Toulouse, Toulouse, France (juillet 2012)
13. Saeys, Y., Inza, I., Larrañaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics 23(19), 2507–2517 (2007)
14. Shafti, L.S., Haya, P.A., García-Herranz, M., Pérez, E.: Evolutionary feature extraction to infer behavioral patterns in ambient intelligence. In: Paternò, F., de Ruyter, B., Markopoulos, P., Santoro, C., van Loenen, E., Luyten, K. (eds.) AmI 2012. LNCS, vol. 7683, pp. 256–271. Springer, Heidelberg (2012)

ARMAN: Agent-based Reputation for Mobile Ad hoc Networks

Guy Guemkam[1,2], Djamel Khadraoui[1], Benjamin Gâteau[1], and Zahia Guessoum[2]

[1] Centre de Recherche Public Henri Tudor, 29 Avenue JFK, 1855 Luxembourg, Luxembourg
{firstname.surname}@tudor.lu

[2] Laboratoire d'informatique de Paris 6, 4 place Jussieu 75252 Paris
{firstname.surname}@lip6.fr

Abstract. Reputation-based trust systems have received large attention as a mechanism to enforce cooperation in decentralized environment such as Mobile ad hoc Networks (MANET). Performance of the MANET results from the cooperation which may be selfish. In such context, avoiding entities that could exhibit malicious behaviour is a challenge. These malicious behaviours include on-off, selfishness, collusion and oscillating. In this paper, we proposed an agent-based reputation approach for MANET that has threefold goals: first it is an enhancement scheme for existing routing protocol in terms of quality of service such as packet delivery. Second, without any prior information, an evaluator agent can form a view about an evaluated agent by requesting and interpreting the reports provided by recommender even when the majority of these reports are unfair. This is done through the similarity of beliefs between agents. Finally the Dempster Shafer theory is used during the integration of information provided by recommender to reduce the uncertainty in trust evaluation.

Keywords: Reputation, trust, similarity, mobile ad hoc networks, Dempster Shafer theory.

1 Introduction

Reputation and trust models play an important role in distributed systems such as Mobile Ad Hoc Networks (MANET). Several distributed solutions have been proposed to enforce security where collaboration among entities is required. Each entity can collaborate to improve the quality of service over the network. For instance a relay entity can refuse to cooperate or deliberately drop the packets. In MANET, relationships among entities arise, evolve and expire on the fly and have a short life span. Indeed, the decentralized nature of such environments implies the cooperation of entities, sometimes in a selfless manner, to achieve a goal. Moreover, ensuring communication among entities in such context is a daunting task. It includes to accurately detect malicious behaviours in the network such as selfishness, which refers to the process of dropping all packets for

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 122–132, 2013.

instance. However, more strategic behaviours have been recognized as threats that downgrade the performance of MANET [13]. These behaviours involve entities that forward packets and begin then to drop all received packets referred as *oscillating behaviour*, even those that collude in order to gain advantage of the network. An entity alternates *on period* where it decides to forward packets with *off period* where it decides to drop packets. This behavior is referred as *on-off*. In this context, the main challenge is to endow any given entity in a MANET with the possibility to accurately detect those behaviours. Reputation-based trust systems have been proposed to cope with these behaviours. In such systems, a value associated to the activity of an entity called *reputation* is used to assess its trustworthiness. We consider that *reputation is a measure that is derived from direct and/or indirect knowledge of earlier interactions if any, and is used to access the level of trust an agent puts into another.* This definition is an extension of the one proposed by [8]. Hence, trust is the probability of the effectiveness that an agent behaves as expected.

Despite the recent developments in reputation-based trust systems in MANET, there is still a need to develop resilient and efficient models that are robust against the most typical attacks such as on-off, collusion (*bad-mouthing* and *ballot-stuffing*), selfishness and oscillating behaviour.

To address this issue, we propose an Agent-based Reputation for Mobile ad Hoc Networks (ARMAN). This solution has a threefold goal: first, it is an enhancement scheme for existing routing protocols in terms of quality of service such as packet delivery. It uses second-hand information even when it is biased and when there is a lack of information about the trustee. Second, without any prior information about a trustee, an ARMAN agent can form a view about this trustee by requesting and interpreting the reports provided by recommender even when the majority of these reports are unfair. This is done through similarity of beliefs between agents. Finally, the Dempster Shafer theory [11] is used during the integration of second-hand information to reduced the uncertainty in trust evaluation.

This paper is organized as follows: Various reputation systems from literature are discussed in Section 2 along with their limitations or vulnerabilities to certain types of attacks. Section 3 introduces our reputation-based trust model which is based on similarity of beliefs and Dempster Sharer theory.The proposed reputation model is evaluated with respect to its different effectiveness against the mentioned threats in Section 4. Conclusion and future research are presented in Section 5.

2 Related Work

Numerous reputation models have been proposed to enforce cooperation such as [15], [16]. However these models need to be analysed by also taking into account their effectiveness against numerous attacks that downgrade the performance of MANET such as collusion, On-Off, oscillating behaviour and selfishness. We also consider models that take into account information coming from other referred as *second-hand information.* A review of existing models for multi-agent systems can be found in [2]. Moreover, in [1], author discusses how important is the robustness in reputation and trust models for on-line systems. These models are viewed as a mean to predict future behaviour of parties involved in a transaction. In [3], a distributed reputation protocol for MANETs called CONFIDANT is proposed. His goal is to make existing systems robust against inaccurate rating provider on one hand, and efficient in detecting misbehaviour on the other hand. The protocol consists of a *Trust Manager, Reputation System, Path Manager, and Monitor.* The Trust Manager controls how second-hand information reports are handled by determining the trustworthiness of a message based on the trust level of the sender. ALARM messages are sent to friends of a node when abnormal behaviour is detected by the node. The validity of these messages is verified by a central trusted entity. Source of information used by the reputation component is first and second-hand information. The Path Manager implements decisions made by the reputation systems. The attack considered in this reputation protocols is false information propagation. Other attacks such as One-shot are not taken into account.

In [5], the authors proposed a Bayesian network approach for modelling trust in multi-agent systems called TRAVOS. In this model, an agent uses two methods to assess the trustworthiness of another based on his direct experiences and from second-hand information. The first method is based on the beta density function used to model direct experiences between two agents named trustor and trustee. TRAVOS assesses each second-hand information individually based on the accuracy of previous information. The model deals with the trustworthiness of others by measuring how likely a witness assessment requester's matches his own assessment. Then a heuristic is used to discount ratings by a factor corresponding to the accuracy of the probability. This approach is based on the hypothesis that the second-hand information given by a provider agent is consistent. This assumption does not consider that an agent may provide misinformation.

The authors in [6] propose a reputation framework for sensor networks to identify misbehaving nodes and inconsistent measures in real time called RFSN. They use two modules to achieve this goal. The first module is used to monitor the actions of other nodes. They distinguished each action into *cooperative* and *non-cooperative.* This module, called *watchdog*, is also used to detect invalid data. Reputation representation, implementation and trust evolution are taken into account through the last module called *reputation system.* This module is also used to detect misbehaving node.

In [7], the authors propose a framework to enhance security in MANET called *RipSec.* It is composed of two major modules called *feedback item* and the *reputation index.* The first module takes the form of opinion coming from others. The second module is used to mitigate threats introduced by misbehaving nodes. This is done by assigning positive assessment to nodes based on their participation to the routing process. Nodes that desire to send data eliminates relay nodes with low reputation during the route establishment. Other attacks such as *bad mouthing*, or *on-off*, are vulnerabilities of this protocol.

In summary, most existing reputation models are able to identify agents that exhibit behaviour such as selfishness and collusion. However, there is no reputation model that considers all the mentioned vulnerabilities. In the next section, we present our reputation approach and its application to MANET.

3 Our Agent-based Reputation for Mobile Ad-hoc Network(ARMAN)

Our model is based on the reputation approach proposed by [5]. In this model trust is seen as the subjective probability by which the model has no resilience on any third-party and all agents decide themselves. It uses both direct information, coming from personal observations of agents, and indirect information coming from other agents. This indirect information represents the synthesis resulting by aggregating observations coming from multiple sources. We consider the following assumptions related to the characteristics of MANETs:

– Our approach does not consider mobility.
– The abnormal behaviour is caused by the agents. However, some abnormality may be caused by several factors such as the unreliability of channels, communication links due to signal noise, etc.

We also consider the trustworthiness associated to the testimonies provided by reporter agents. We make this distinction because most of the models are based on the assumption that an agent cooperative in packet delivery will be trustworthy in providing testimonies and vice versa. Moreover, each ARMAN's agent follows the steps depicted in Figure 1.

At the beginning, the *Monitoring* step is performed by an agent to detect incoming transactions (a transaction regroups every interaction between agents, i.e. message forwarding) from other agents. This *Monitoring* phase follows a kind of *watchdog* methodology where each agent monitors the behaviour of neighbouring nodes in a promiscuous mode to overhear whether a packet has been forwarded or not. When an agent finishes the monitoring phase, it starts the decision phase. Depending on the received transaction and the emitted status i.e trustworthiness, that agent can require entering several times in the computation phase to compute reputation values (direct/indirect reputation and trustworthiness) needed to make a decision. Regulated by these values this agent can make the choice to perform actions. Finally when the action phase is completed the agent returns to his initial monitoring phase.

3.1 Reputation Computation

The reputation method is performed in three parts:

– The first part concerns reputation of agents obtained through direct observations between a *truster* and *trustees* as in [5];

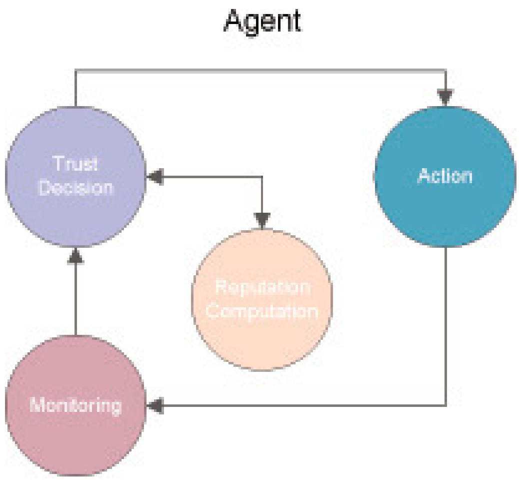


Fig. 1. Agents Step

- The second part concerns reputation obtained through second-hand information provided by recommender agents that belong to a set of neighbours;
- The last part concerns the integration of second-hand information using the Dempster-Shafer theory.

Given a *truster* agent A and *trustee* C, a number of positive and negative observations respectively denoted by α and β. The direct reputation value is the mean of the *Beta* distribution between A and C. It is calculated at time t as the expectation value of $Beta(\alpha, \beta)$ denoted E:

$$DR^t_{A,C} = E_{A,C}(Beta(\alpha^t, \beta^t)) = \frac{\alpha^t + 1}{\alpha^t + \beta^t + 2} \tag{1}$$

where α^t and β^t represent the numbers of positive and negative observations respectively a time t. Given $\alpha^t = \sum_{i=1}^{t} p_i \gamma^{t-i}$, $\beta^t = \sum_{i=1}^{t} n_i \gamma^{t-i}$, γ representing the forgetting factor, and p_i stands for positive observations at time i, n_i negative observations at time i. This method is used by Agent A to estimate the reputation of Agent C. Moreover, given a set of neighbouring agents N_i, the overall reputation(OR) of an agent is calculated based on his personal observations and also on *trustable* observations coming from others using equation 2. These observations are considered as *indirect reputation(IR)*.

$$OR^t_{AC} = \delta DR_{AC} + \frac{(1-\delta)\sum_{i=1}^{n} \lambda_{Ai} IR_{iC}}{\sum_{i=1}^{n} \lambda_{Ai}} \tag{2}$$

In this Equation 2, λ_{Ai} is the trustworthiness of the agent i view by agent A in providing accurate recommendation, δ is used to give more or less important weight to direct and indirect reputation.

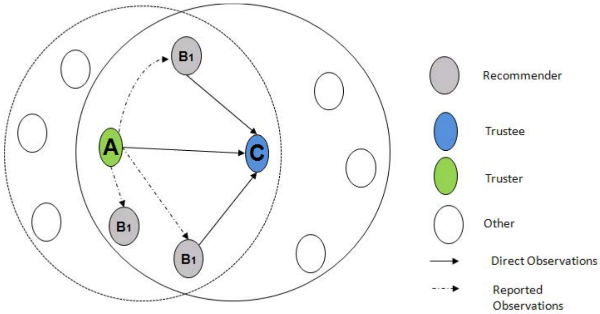


Fig. 2. Agent Assessment

3.2 Detection of Maliciousness

Considering that an agent may be good at forwarding messages and at the same time provides false observations, we use *similarity view* to deal with this issue. This *similarity view* is based on the assumption that if two agents observe an event in the same way, they have similar views as underlined by [12]. Given two agents, A and B_1 as shown in the Figure 2. Let A be the *truster* agent, let B_1 be a recommender agent located in his neighbourhood and let $Set(A, B_1) = \{C, D, E, ..., N\}$ be a set of agents they have interacted together. The *similarity view* between A and B_1 is calculated as follows:

$$sim^t(A, B_1) = 1 - dis^t(A, B_1) \tag{3}$$

$$dis^t(A, B_1) = \sqrt{\sum_{i \in Set(A,B_1)} (DR_{A-i} - DR_{B_1-i})^2} \tag{4}$$

So, the metric used to assess the trustworthiness of agents is based on the ratio of packet delivery, and also that there is a failure delivery due to time delay. Observations coming from other recommender agents are accepted if their *similarity view* with the truster is greater than a threshold value (ν). Otherwise their observations are rejected.

3.3 Trust Decision

We represent trust relation among agents by a real number within the interval [0,1]. Trust value is obtained using the overall reputation in equation 2 as follows.

$$T^t_{A,C} = OR^t_{AC} \tag{5}$$

The lowest value represents the fact that agent A trusts agent C and the highest represents the fact that A distrusts C.

In order to integrated indirect reputation to compute trust of an agent and to reduce the uncertainty of trust evaluation we use the Dempster Shafer theory instead of weighted average one. The next lines describes how this theory is applied.

Assuming a frame of discernment $\theta = \{U(Untrustable), T(Trustable)\}$, the power set of θ is $2^\theta = \{\varnothing, \{T\}, \{T,U\}, \{U\}\}$, a trust function called bel_A, which satisfies $bel_A(\varnothing) = 0, bel_A(\theta) = 1$. This belief function measure the total belief in the proposition represented by each subset of θ and represents basic probability assignment (bpa)or belief masse. Then $m_A(T)$ represents the part of the degree of belief parts that agent has on the suggestion. Indeed, given observations coming from two distinct sources m_{B_1}, m_{B_2}. The fusion of information from different sources is performed using the law of combinations of Dempster and Shafer also called textitorthogonal sum denoted by: $\forall^T \in 2^\theta$,m(T)=$m_{B_1}(T) \bigoplus m_{B_2}(T)$. Given $K = \sum_{X \cap Y = \varnothing} m_{B_1}(X) \bullet m_{B_2}(Y) < 0$.

$$m_A(T) = \frac{\sum_{X \cap Y = T} m_{B_1}(X) \bullet m_{B_2}(Y)}{1 - K} \tag{6}$$

Note that, this *orthogonal sum* is associative and commutative, indeed we can combine more than two evidence pairwise in any order. The combination of evidence at agent A given by agents B_1 andB_2 is illustrated in Table 1. The uncertainty in information from B_1 and B_2 are $Bel_{B_1}(\{U\}) = 0.3$, $Bel_{B_2}(\{U\}) = 0.4$ respectively. The probabilities coming from reputation process are combined using 6. Therefore the beliefs in the *trustable(T)*,*untrusted(U)* and the *uncertainty* are calculated as follows: $bel_A(\{T\}) = m_A(\{T\}) = m_{B_1} \bigoplus m_{B_2}(\{T\}) = 0.88$, $m_{B_1} \bigoplus m_{B_2}(\{U\}) = 0$ and $m_{B_1} \bigoplus m_{B_2}(\{T,U\}) = 0.12$. Given the evidence provided by the two reporters B_1 and B_2 to agent A, the decision made is to consider that the targeted agents is trustworthy. Following this, trust that A has on C follows the decision process described in 7.

$$\begin{cases} m_{A,C}(\{T\}) > m_{A,C}(\{T,U\}) \\ m_{A,C}(\{T\}) = \nu(threshold) \end{cases} \tag{7}$$

Table 1. DS evidence combination at Agent A

B_2 \ B_1	{T}:06	{U}:0	{T,U}:0.4
{T}:0.7	0.42	0.0	0.28
{U}:0	0.0	0.0	0.0
{T,U}:0.3	0.18	0.0	0.12

4 Experimental Evaluation

To validate our proposal, we first implement ARMAN model in JAVA and three models from the literature : CONFIDANT, RFSN and TRIP [9]. We also simulate random decentralized MAS Network in a graphic environment as depicted in Figure 3.

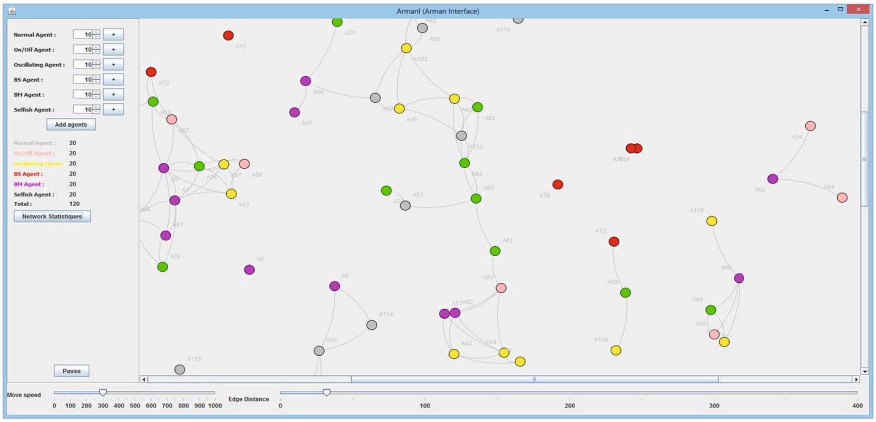


Fig. 3. Simulation framework

4.1 Experiment Environment

To represent the dynamics of the network, agents have the ability to move around the graph at a fixed speed in a completely random behaviour (agent select a point in the graph and reach it) with no possible collision between them. During their travel, agents can receive transactions and decide to forward them depending on their environment (trustable neighbours), the sender (trustable sender) and also the content of the transaction (TTL, recipient, ...). Furthermore, the exchange of reputation information may increase communication overhead. Indeed, the comparison with the model mentioned in this paper is left for future work. Although [17] shows that each agent in a network of N agents requires $O(N)$ running time to update its reputation table. The simulation parameters are listed in table 2.

Detection Technique

To obtain statistic of detection during the simulation we have added into the detection methods of each model a test method that verifies if the detected agent

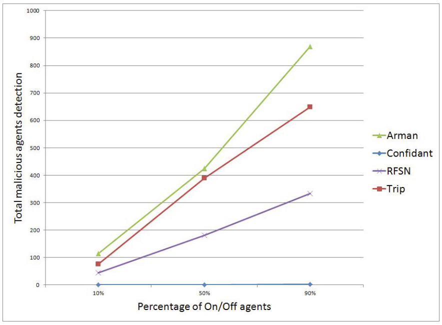
Table 2. Simulation environment parameters

Parameter	**Value**
Dimension of space	600x600px
Maximum edge distance	25px
Agent movement speed	1px per 300msec (3,3px per second)
Number of Agents	100
Simulation time	120 secs
Total of transaction	2400 (24 transactions/ agent, one transaction/5sec)
Percentage of malicious agents	10% - 50% - 90%

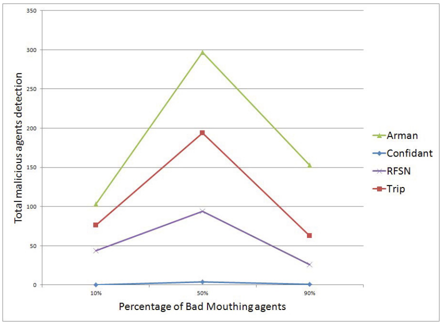
is really a malicious one by checking his role. If he is playing a malicious role we count it as a positive detection, if not as a negative detection.

4.2 Experiments and Results

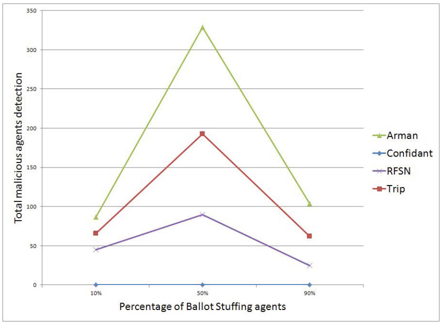
We have performed several scenarios to check the robustness against several attacks. Figure 4 shows the effectiveness against 10%,50% - 90% of malicious agents. The total number of malicious agents detected per model for positive detection is used to assess the effectiveness of a model. The mobility of the network



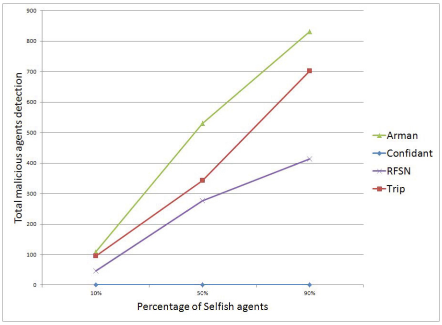
Effectiveness against On-Off



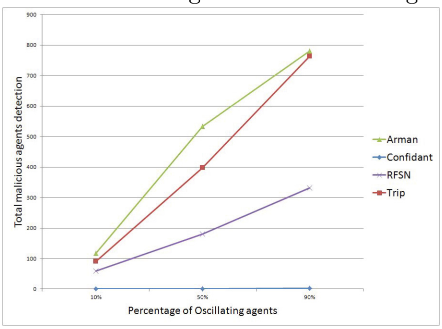
Effectiveness against Bad-Mouthing



Effectiveness against Ballot-Stuffing



Effectiveness against Selfishness



Effectiveness against Oscillating

Fig. 4. Simulation Results with ARMAN, CONFIDANT, RFSN, TRIP

influences these results. For instance a network that is not fully interconnected as in the Figure 3 will give poorer results. Therefore, the figure that depicts the effectiveness against bad-mouthing shows that the maximum numbers of malicious agents detected is *300* for 50 malicious agents. The value *300* represents the total number of malicious agents detected during the simulation. For instance, for m malicious agents and N total agents, we can have $m \times (N - 1)$ detections. Given that a malicious agent can detect another malicious agent. Indeed, the number of detection per model grows with the percentage of malicious behaviours except Bad-Mouthing and Ballot-Stuffing. In such cases, the collusive nature of the attack decreases this percentage. This is due to the fact that each Bad-Mouthing or Ballot-Stuffing agent does not detect his colleague as malicious. Therefore, as the number of malicious agents exceeds 50%, the detection of their maliciousness decreases. Detection of On-Off agent is well performed by ARMAN and Trip

with proportional evolution of the detection count per model. RFSN detection count is less than TRIP and ARMAN in the different scenarios. Selfish agent is well detected by all models. ARMAN outperforms other models in the case of oscillating behaviour between 10% and 90% of malicious. When the percentage of malicious behaviours reaches 90%, TRIP and ARMAN have identical results. In the case of ballot-stuffing, ARMAN model has better results than the other models.

5 Conclusion and Future Work

This paper presented a reputation based trust model called ARMAN. This model combines techniques from statistics and Dempster-Shafer theory for robust detection of malicious behaviours as well as false recommendations. Unlike existing schemes [6],[3],[9], ARMAN considers the experiences of the truster with the trutee during the reputation calculation as well as the trustworthiness of an agent as recommender. We experimented with various types of malicious behaviours such as collusion, oscillation, on-off and selfishness. The comparison with similar schemes such as CONFIDANT, RFSN, and TRIP shows that our approach outperforms these models. In the future, we plan to examine the overhead in terms of information exchange to reach the reputation value.

References

1. Jøsang, A.: Robustness of Trust and Reputation Systems: Does It Matter? In: Dimitrakos, T., Moona, R., Patel, D., McKnight, D.H. (eds.) IFIPTM 2012. IFIP AICT, vol. 374, pp. 253–262. Springer, Heidelberg (2012)
2. Sabater, J., Sierra, C.: Review on computational trust and reputation models. Artificial Intelligence Review 24(1), 33–60 (2005)
3. Buchegger, S., Boudec, J.Y.L.: A robust reputation system for peer- to-peer and mobile ad-hoc networks. In: Workshop on the Economics of Peer-to-Peer Systems (P2PEcon 2004), Cambridge (2004)

4. Jøsang, A., Ismail, R.: The beta reputation system. In: Proceedings of the 15th Bled Electronic Commerce Conference, pp. 41–55 (2002)
5. Teacy, W., Patel, J., Jennings, N., Luck, M.: TRAVOS: Trust and reputation in the context of inaccurate information sources. Autonomous Agents and Multi-Agent Systems 12(2), 183–198 (2006)
6. Ganeriwal, S., Balzano, L.K., Srivastava, M.B.: Reputation-based framework for high integrity sensor networks. ACM Trans. Sen. Netw. (2008)
7. Lacey, T.H., Mills, R.F., Mullins, B.E., Raines, R.A., Oxley, M.E., Rogers, S.K.: RIPsec using reputation-based multilayer security to protect MANETs. Computers & Security (2012)
8. Aberer, K., Despotovic, Z.: Managing Trust in a Peer-2-Peer Information System. In: Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM 2001), pp. 310–317 (2001)
9. Mármol, F.G., Pérez, G.M.: TRIP, a trust and reputation infrastructure-based proposal for vehicular ad hoc networks. Journal of Network and Computer Applications 35(3), 934–941 (2012)
10. Vercouter, L., Jamont, J.: Lightweight Trusted Routing for Wireless Sensor Networks. In: Demazeau, Y., Pechoucek, M., Corchado, J.M., Pérez, J.B. (eds.) Adv. on Prac. Appl. of Agents and Mult. Sys., April 6-8. AISC, vol. 88, pp. 87–96. Springer, Heidelberg (2011)
11. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press (1976)
12. Guemkam, G., Feltus, C., Schmitt, P., Bonhomme, C., Khadraoui, D., Guessoum, Z.: Reputation Based Dynamic Responsibility to Agent Assignement for Critical Infrastructure. In: Proceedings of the 2011 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2011, Lyon, France, August 22-27 (2011)
13. Gagandeep, Aashima, Kumar, P.: Analysis of Different Security Attacks in MANETs on Protocol Stack A-Review. International Journal of Engineering and Advanced Technology (IJEAT) 1(5) (June 2012) ISSN: 2249 8958
14. Mundinger, J., Le Boudec, J.: Analysis of a Reputation System for Mobile Ad-Hoc Networks with Liars. Performance Evaluation 65(3-4), 212–226 (2008)
15. Zhao, L., Delgado-Frias, J.G.: MARS: Misbehavior Detection in Ad Hoc Networks. In: IEEE Global Telecommunications Conference (GLOBECOM 2007), pp. 941–945 (2007)
16. Bansal, S., Baker, M.: Observation-based Cooperation Enforcement in Ad Hoc Networks, Stanford University Technical Report (July 2003)
17. Chang, B.-J., Kuo, S.-L.: Markov Chain Trust Model for Trust-Value Analysis and Key Management in Distributed Multicast MANETs. IEEE Transactions on Vehicular Technology 58, 1846–1863 (2009)

Decentralized Intelligent Real World Embedded Systems: A Tool to Tune Design and Deployment

Jean-Paul Jamont, Michel Occello, and Eduardo Mendes

Université de Grenoble-Alpes, LCIS, 26000 Valence, France
firstname.lastname@lcis.grenoble-inp.fr
http://lcis.grenoble-inp.fr

Abstract. This paper presents an approach and a tool, called MASH, to design of real world decentralized intelligent systems. MASH enables the simulation of distributed systems including virtual and real world embedded nodes according to realistic physical models. We present the key features of this tool and its architecture.

Keywords: embedded MAS, deployment, simulation.

1 Introduction

Context. More and more real world intelligent systems consist of lots of small interconnected devices which interact together. These components must be small, inexpensive, and therefore as simple as possible. Designing such solutions requires to take into account strong criteria like energy consumption, low CPU power, low memories etc.

In the literature, we can consider two types of approach to design such systems. *Decentralized approaches* as multiagent systems (MAS) and other distributed artificial intelligence solutions. Systems are seen as sets of interacting autonomous entities (agents) reasoning about a partial description of their environment. *Centralized solutions* as traditional automation based solutions. A global model of the environment is defined and maintained. The decisional process uses this model to decide and to act.

In the first case, interesting applicative properties (like stability) are difficult to prove because of the decentralized and distributed nature of the system. The possible number of exchanged messages due to the cooperation process between the distributed entities can be a problem too. In the second case, obtaining a realistic global model can be expensive in term of computation and in term of information transport.

We focus on *decentralized* intelligent systems. Designing software for such systems is a difficult task due to the inherent complexity at both conceptual and implementation levels. Systems can be observed at both an individual level and a social level. The *individual level* focuses on capacities, knowledge and goals of entities. The *social level* focuses on global aspects of the whole system, external

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 133–144, 2013.

expression of interaction and cooperation situations. From local individual interactions can emerge behaviors at the system level which are difficult to predict.

Problem. Designing and deploying a decentralized intelligent systems requires to simulate the behavior of the whole designed system. Because of the complexity of applicative problems, simulation tools must help the designer to investigate important features of local control strategies and to inspect their effects on the global behavior. The precision of the models used in the simulation (physical environment, energy consumption, wave propagation...) has a significant impact on the quality of the whole solution that will be really deployed in the real world. To support the system deployment, we must simulate decentralized intelligent systems constituted by virtual (simulated) parts and embedded parts really deployed in the real world.

Such a tool must support an important variety of models coming from different fields. These models are mainly *organizational models* (self-organization process, hierarchy management etc.), *interaction models* (contract net protocol, recruiting interaction protocol etc.), *physical environment models* (including different classes of application dependent models as wave propagation models, thermal dissipation model, fluid flow etc.) and *user models* (implementing different types of user's specific needs dependent behavior).

Contribution. MASH (MultiAgent Software Hardware simulator) tries to meet these requirements. This tool is used according to the following approach (figure 1). This approach is involved in a more complete system design methodology called DIAMOND [11] which is not presented here.

When the simulated solution meets the requirements it is necessary to embed the solution in the real world devices. A specific effort is needed to tune algorithms in order to fit resources of devices. Algorithms must be simplified to accommodate, for example, memory limitations, reduced computation capacities etc. Deviations of the global behavior may result from these modifications. Tools

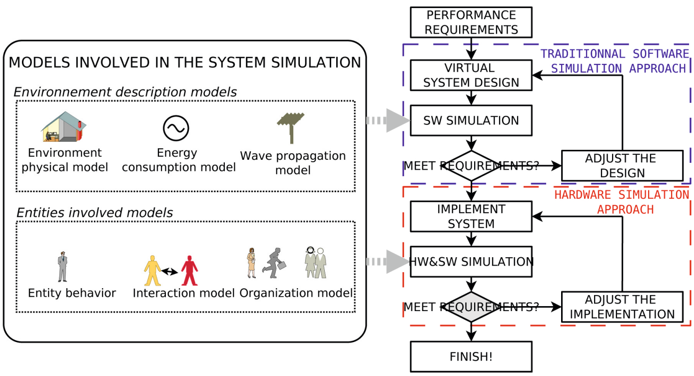


Fig. 1. Using MASH to design and to deploy real world MAS

must help to control effects of local changes in the behavior of entities on the global behavior.

This paper gives an insight to our pragmatic design approach and focuses on its associated tool. Applications are given in a companion demo paper within the same proceedings [12].

The structure of the paper is as follows: Section §2 introduces the MASH architecture. In section §3 we focus on the key features of our tool : the virtual/real world mixed society simulation and the use of realistic physical models. For each of these key features the related works are exposed.

2 An Introduction to MASH Architecture

2.1 Preliminary Definitions

Embedded MAS include different types of agents:

– **Embedded agents** are agents embedded in the real world which can perceive and act on physical environment. They are often constituted by a software part and a hardware part. Soccer robots, autonomous vehicle, intelligent sensors are embedded agents.
– **Software agents** are traditional agents which can perceive and acts on virtual environment.
– **Virtual agents** are a type of software agents used to simulate the behavior of real world embedded agents.
– **Avatar agents** represent embedded agents in virtual societies. They enable embedded agents to interact with software agents. They link the simulated MAS with behaviors computed on physical devices.

Virtual environments are environments in which parameter values are estimated/computed from their physical models. Parameter values of physical environments are acquired by sensors and they are modified by effectors.

We call **virtual society** the set of software agents of the multiagent system. Embedded agents are not taken into account contrary to their avatars.

We call **instruction** a method call on an agent (or an object) at a specified date (date,<object_identifier>.<method_name>(<param1>,...)) . A **scenario** consists in a set of instructions.

2.2 Architecture Overview

MASH enables involving software simulation (involving software agents), the hardware simulation (only real world embedded agents operating on a simulated environment) and the virtual/real world hybrid simulations (involving software agents and real world embedded agents operating on a simulated/real world mixed environment).

In figure 2, we can see in the background the main windows which allow to view the system according to customized criteria (defined by the designer). It is possible to spy agents i.e. to inspect their internal states and the history of events. The inspected agent is Agent2 which is a real world agent. Among the various events, we can see the bytes received by this agent and their translation into logical messages.

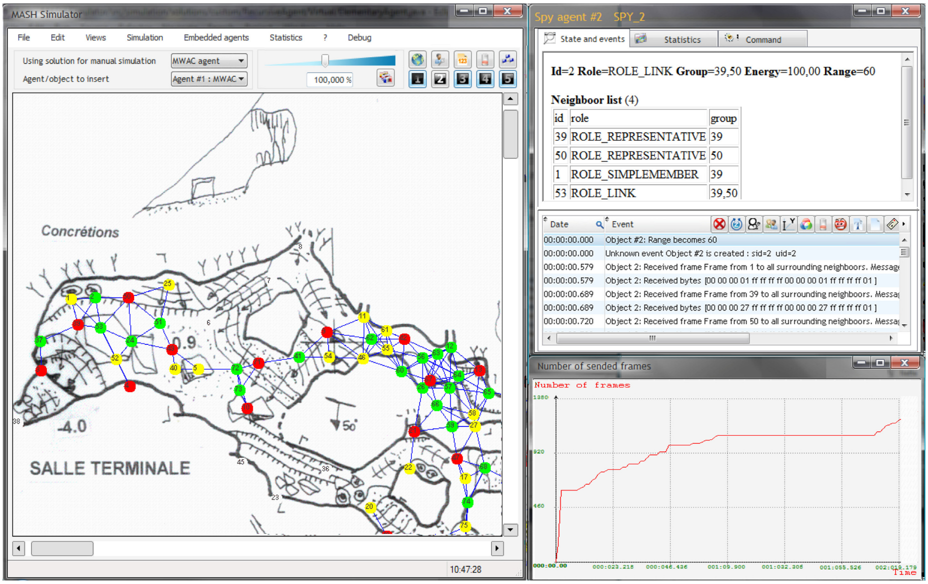


Fig. 2. The MASH main windows

Virtual agents and real world embedded agents are abstracted by an *Individual agent manager* (figure 3). The *Individual Agent Manager* enables the integration in the simulation of virtual and real world agents. Each agent possesses its own model and its own architecture. An agent can be implemented by a software agent (as a java class) or its behavior can be computed in a real world embedded agent. In this case, an avatar translates the logical call of methods and exchanged messages to its wrapped embedded agent. Physical connections are implemented according to a given bit specification (defined by user). The avatar allows to give a graphical representation of the real world embedded agent in the MAS graphical representation.

The *Behavior component* is the applicative component. It simulates the execution of software on a single node. It processes messages received from other agents. The agent's decision cycle dedicated to the functional aspect of the application is implemented here.

Agents interact together and with the environment through the *society manager* and the *Environment manager*. The *Society Manager* defines the locality of an agent. In other words, it enables an agent (1) to identify its neighborhood i.e. the agents that can physically receive the messages it transmits, (2) to

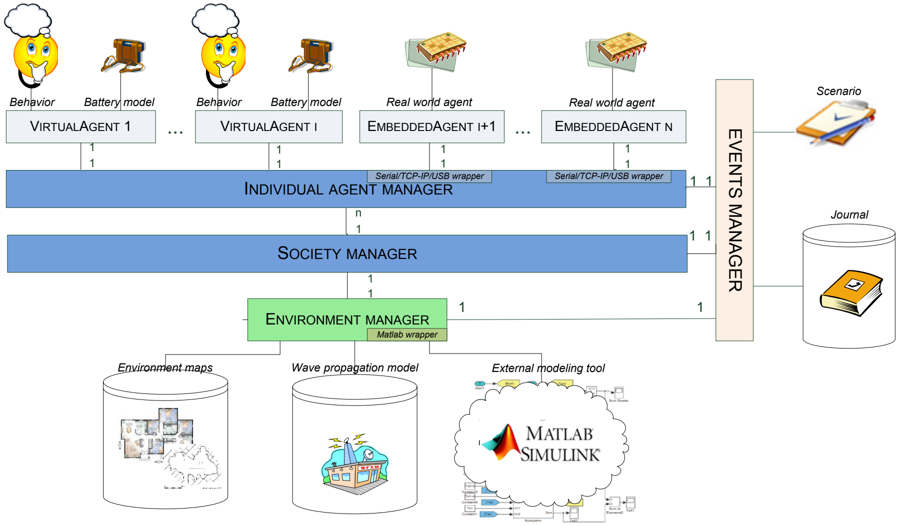


Fig. 3. Simplified architecture of the MASH simulator

access the environment values that its sensors can measure and (3) to act on the environment (to modify environment parameters) with their effectors.

The *Environment manager* computes different physical models to allow a realistic simulation. When an agent wants to capture an environment value , this component decides firstly the accessibility of this data by the agent (Does the agent have the appropriate sensor? Does the agent is in the good geographical area?). Secondly, it returns the value or it throws an exception.

Concerning the *scenario* processing, the simulator uses the reflection API of java which allows to examine or to modify the runtime behavior of Java applications. MASH uses it to examine properties of agents, including their declaration and their contents. Main advantages of use of this API is the possibility, when a scenario is running, to call dynamically user defined methods/services without any additional declaration of possible actions.

3 Key Features

In this part, we focus on main particularities of MASH. For each of them, we present a quick review of related works and give an insight of its implementation in MASH.

3.1 Simulation of Real World/Virtual Societies

Software actors and hardware actors mixed systems are increasingly common, but relatively few tools enable their development.

Related Works. Some works support simulation of systems including hardware entities and software entities. These works often belong to the wireless networks field. These simulators [14,7,22,23] are often specialized in the study of a specific type of model (battery discharge model, wave propagation model...). We identified no contribution in the field of decentralized artificial intelligence which enables to involve software agents and real world embedded agents in a same simulation. In [24], authors make a virtual discrete environment from the physical observations to plan actions. In [9], the simulator acquires data from sensors to obtain more realistic virtual simulations.

Concerning MASH. In our tool, all real world embedded agents have an avatar in the simulated society. These avatars are managed by the *Individual agent manager*. Their main role is to translate logical messages into physical ones and vice versa. They can be used to compare real world agent behaviours to the behaviour of their virtual implementation.

As an instance, figure 4 shows the expected role and the measured role of an hardware agent included in the simulator. At `t=152.45s`, we can see that the real world agent requires more time to choose its next role (the CPU clock is lower than the virtual agent clock). At `t+dt`, we can see that the real world agent has chosen an unexpected role (simple member). Reason is that a more important amount of time is needed to analyze the received messages.

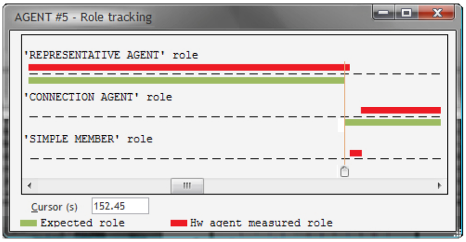


Fig. 4. Tracking an agent role

3.2 Simulation Using Realistic Physical Models

The majority of distributed intelligence systems concerns only virtual applications. For example, in the context of home automation, simulators belonging to the distributed intelligence field focus on software problems like interaction protocols used to negotiate users' needs, decision making and distributed problem solving to adapt these needs to energy limitation [2,1], data-mining to match the specific situation to a previously observed one [6].

The associate tools like [3,18,13,17] are not suitable for the design of real world systems especially because physical laws of these environments are indeed reduced to their simplest expression.

Environment Models. The environment models in which agents evolve are often reduced to their simplest form (in most of works of ambient artificial intelligence). However their impact on simulation results is very important. We present related works and a use of more realistic models in MASH.

Related Works. The most interesting solutions are those that use the well known tools Labview [5] or Matlab[16]. In [20,4], authors use Labview to implement a MAS. In their simulations an agent is implemented as a virtual instrument. Of course, by this way authors lose the advantages of multiagent simulators (large scale simulation, multiagent specific models...). Very recently, in another context a matlab/simulink multiagent toolkit for distributed networked fault tolerant control systems has been proposed [15]. In [21], MACSimJX an extension of Jade is proposed to enable interaction between Simulink and this tool.

Concerning MASH. Matlab is undoubtedly the most suitable tool to model physical systems even if the creation of a model requires to have serious knowledge about the physical law that we want to model.

A Simulink model is presented as block diagram. Such diagrams enable to solve set of algebraic equations and ordinary differential equations. This block organization enables easy reuse of already developed blocks and allows to have a better comprehension of the whole model.

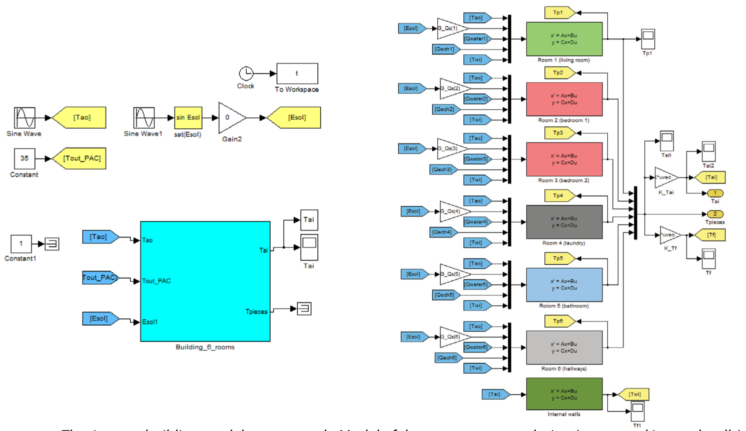
In [10], we propose to model[1] temperature evolution of each room of a building. The temperature of a room depends, on one hand, of elements that are only dependent on that particular room and of external parameters, and on the other hand of shared components, such as walls and doors, with other rooms. Consequently, we propose the following methodology: making one model for each piece with its own components (air, floor, ceiling, external walls, windows, internal heating sources...) and one model for each linking component (walls and doors between rooms, floor/ceiling for multilevel building...). Mainly, the first models (room model) are connected using the second models (internal building walls). This methodology permits to connect and disconnect the rooms very efficiently without having to rewrite the equations of the different components. One has first to define the components (rooms and internal walls) connectivity and second the physical parameters of each component.

The resulting modular model is implemented into MatLab/Simulink. The model enables to change the building configuration easily. Some parts of the model are exposed in figure 5. We find on this figure the entire model of the 6 rooms building that we simulate (fig. 5a) and how we compute the temperature of the rooms and of the internal walls composing the building according equations described in [10] that are in state space form (fig. 5b).

To use a physical model defined with Matlab/Simulink, MASH can interact with this tool by two ways : using a client/server approach or by files exchange.

Concerning the first solution, the Matlab Real-Time Workshop is able to provide the code to implement both a server and a client. We use only its server code generation. MASH supplies the client code to access to the physical model

[1] This model results from our collaboration with automation researchers.



a. The 6 rooms building model b. Model of the temperature evolution (rooms and internal walls)

Fig. 5. Environment physical model : Part of the Matlab block model

parameters. Because this solution requires to buy a specific module, we preferred to develop a little Matlab script which enables a double queue file communication. These text files (default names: `data.in` (inputs of the physical model i.e. outputs of MASH) — `data.out` (outputs of the physical model i.e. inputs of MASH)) enables to share parameters and measures. An example of these exchanged file is given in figure 7.

Then, MASH therefore uses a very simple algorithm (alg. 1) to put the update data relating to the environment in which agents evolve.

Algorithm 1. Integrating Matlab/Simulink computed values

---

repeat
  Read environment values in `data.out.xml`
  Update MASH virtual environment
  Agents act
  Write controllable values by agents in `data.in.xml`
until simulation.isRunning()

Energy Consumption Model. Energy consumption models are very important to obtain realistic results from the energy efficiency point of view. It is necessary for the software simulated agent to include the energetic consumption in the simulation because all embedded agents must integrate the energy point of view in their reasoning.

Related Works. The battery model simulates the capacity and the lifetime of the agent energy source. It is difficult to define a universal model because the battery behaviour strongly depends on the material used to build the agents. For an embedded agent, one of its main goals is to increase as much as possible

```
<?xml version="1.0">
<! -- Written on 24-Aug-2011 10:52:41 using the XML Toolbox for Matlab -- >
<root>
  <data1>
    <name>simulated_time< /name>
    <type>float< /type>
    <value>30< /value>
    <unit>day< /unit>
    <description>Simulation time : day< /description>
     <direction>out< /direction>
  < /data1>
   ...
  <data17>
    <name>Tai_03_03< /name>
    <type>float< /type>
    <value>32.678271140525< /value>
    <unit>celcius< /unit>
    <description>Ambient air temperature < /description>
    <location>room 3< /location>
    <direction>out< /direction>
  < /data17>
   ...
< /root>
```

Fig. 6. Example of XML exchanged files between Matlab/Simulink and MASH

the lifetime of its energy storage. One of the most simplest models of battery is the linear model. Other models are described in [19,8].

In a lot of simulators of decentralized intelligent systems, energy consumption models are not really available. Considering the large variety of batteries existing in the real world, it will be interesting to supply several models.

Concerning MASH. We defined an open structure which allows to implement several models. The more simple one implemented in MASH consists in defining the battery as a linear storage of current. The remaining capacity C after operation of time td can be expressed by the following equation where C' is the previous capacity and $I(t)$ is the instantaneous current used by the hardware at time t : $C = C' - \int_{t=t_0}^{t_0+t_d} I(t)dt$. The designer must tune the values involved in this model. For instance, we can define the consumed current depending on some states: `radio emission` (8.1 mA), `radio reception` (7.0 mA), `cpu active mode` (2.0 mA), `cpu sleeping mode` (1.9 mA). A designer can add new states to model others current consumptions (sensing consumptions, acting consumptions...).

Wave Propagation Model. If for most applications, the use of a particular model of wave propagation is not essential, it may be necessary when the system environment is disturbed non-uniformly. We present here the use of such models in MASH.

Related Works. In tools specialized on wireless network simulation like NS2, numerous models are available. These models are dedicated to specific environment (cities, home indoor...). We can tune some parameters like the bit rate, the bit error rate and so on. In a lot of distributed intelligence simulator, these models don't really exist. Authors consider an arbitrary range or that the environment is fully covered by access points. Associated energy consumption models are often poorly understood by software specialists. An example of false assumption often found in papers is that a system (as a WSN) spends more energy during transmissions rather than during receptions.

Concerning MASH. The wave propagation model can be modelled as a part of the Matlab environment model. Even that, to simplify reuse of such a model, we separate this specific model. So it is possible to have two concurrent Matlab simulation (one for modelling the wave propagation and one for others physical phenomena like heat transfer).

If a so advanced model is not necessary, MASH offers a simple model implementing the Friis free space transmission equation (used in telecommunications engineering) which does not need to use Matlab. The wave propagation model is implemented like circular wave propagation through the 2-dimensional grid. MASH estimates the received power measured by a receiver agent Pr when a sender agent sends the message with transmitter power Ps. Considering the receiver signal strength, we can estimate the probability of a good reception of the message by the receiver agent. Estimating Sr requires to know the geographical position of the sender and of the possible receivers. These positions are stored in the environment map and not in the agent because in a lot of applications, embedded systems cannot know their positions. From these positions, we can identify the different media crossed by the signal (air, water, wall etc.) during its propagation.

4 Conclusion

This paper presents an approach to design and deploy real world systems based on decentralized intelligence based on MAS. We presented MASH a tool supporting this approach.

Building such systems imposes to design and to tune embedded agents. MASH enables using both simulated and really embedded agents in a same system. One of its major benefits is so the possibility to control possible deviations obtained at the system level from the individual level behaviour modification. A second ability of MASH is it supply realistic physical models for agent environment. Coupling virtual agents/embedded agents with physical models leads to a realistic development tool, bringin closer MAS and physical systems.

Currently MASH supports simulation of systems incorporating populations of 600 to 900 agents (depending on the agent architecture complexity). So the next main improvement of MASH consists in enabling the society manager to distribute individual agent manager on clusters of machines.

References

1. Abras, S., Ploix, S., Pesty, S., Jacomino, M.: A multi-agent home automation system for power management. In: Proc. of the 3rd Int. Conf. on Informatics in Control, Automation and Robotics, Intelligent Control Systems and Optimization, pp. 3–8. INSTICC Press (2006)

2. Begg, R., Hassan, R.: Artificial neural networks in smart homes. In: Augusto, J.C., Nugent, C.D. (eds.) Designing Smart Homes. LNCS (LNAI), vol. 4008, pp. 146–164. Springer, Heidelberg (2006)
3. De Carolis, B., Cozzolongo, G., Pizzutilo, S., Plantamura, V.L.: Agent-based home simulation and control. In: Hacid, M.-S., Murray, N.V., Raś, Z.W., Tsumoto, S. (eds.) ISMIS 2005. LNCS (LNAI), vol. 3488, pp. 404–412. Springer, Heidelberg (2005)
4. Conte, G., Scaradozzi, D., Perdon, A., Morganti, G.: Mas theory for resource management in home automation systems. Journal of Physical Agents 3, 15–19 (2009)
5. Fairweather, I., Brumfield, A.: LabVIEW: A Developer's Guide to Real World Integration. Taylor and Francis (2011)
6. Galton, A.: Causal reasoning for alert generation in smart homes. In: Augusto, J.C., Nugent, C.D. (eds.) Designing Smart Homes. LNCS (LNAI), vol. 4008, pp. 57–70. Springer, Heidelberg (2006)
7. Girod, L., Elson, J., Cerpa, A., Stathopoulos, T., Ramanathan, N., Estrin, D.: Emstar: A software environment for developing and deploying wireless sensor networks. In: Proc. of the USENIX Annual Tech. Conf., pp. 283–296. USENIX (2004)
8. Handy, M., Timmermann, D.: Simulation of mobile wsn with accurate modelling of non-linear battery effects. In: Applied Simulation and Modelling. Acta Press (2003)
9. Hassaine, F., Moulton, R., Fink, C.: Composing a high fidelity hla federation for littoral operations. In: Symp. on Applied Computing, pp. 2087–2092. ACM (2009)
10. Jamont, J.P., Mendes, E., Occello, M.: A framework to simulate and support the design of distributed automation and decentralized control systems: application to control of indoor building comfort. In: Proc. of the IEEE Symp. on Computational Intelligence in Control and Automation, pp. 80–87. IEEE (2011)
11. Jamont, J.P., Occello, M.: A multiagent method to design hardware/software collaborative systems. In: Proceedings of the 12th International Conference on Computer Supported Cooperative Work in Design, pp. 361–366. IEEE (2008)
12. Jamont, J.-P., Occello, M.: Using mash in the context of the design of embedded multiagent system. In: Demazeau, Y., Ishida, T., Corchado, J.M., Bajo, J. (eds.) PAAMS 2013. LNCS, vol. 7879, pp. 283–286. Springer, Heidelberg (2013)
13. Kim, I., Park, H., Noh, B., Lee, Y., Lee, S., Lee, H.: Design and implementation of context-awareness simulation toolkit for context learning. In: IEEE Conf. on Sensor Networks, Ubiquitous, and Trustworthy Computing, pp. 96–103 (2006)
14. Levis, P., Lee, N., Welsh, M., Culler, D.: Tossim: accurate and scalable simulation of entire tinyos application. In: Proc. of the Int. Conf. on Embedded Networked Sensor Systems, pp. 126–137. ACM (2003)
15. Mendes, M., Santos, B., da Costa, J.S.: A matlab/simulink multi-agent toolkit for distributed networked fault tolerant control systems. In: Proc. of the 7th IFAC Symp. on Fault Detection, Supervision and Safety of Technical Processes (2010)
16. Moore, H.: MATLAB for engineers. ESource–the Prentice Hall engineering source. Prentice Hall (2001)
17. Nguyen, T.V., Nguyen, H.A., Choi, D.: Development of a context aware virtual smart home simulator. CoRR 1007.1274 (2010)
18. O'Neill, E., Klepal, M., Lewis, D., O'Donnell, T., O'Sullivan, D., Pesch, D.: A testbed for evaluating human interaction with ubiquitous computing environments. In: Proc. of the 1st Int. Conf. on Testbeds, Research Infrastructures for the Development of Networks and Communities, pp. 60–69. IEEE Computer Society (2005)
19. Park, S., Savvides, A., Srivastava, M.B.: Simulating networks of wireless sensors. In: Proc. of the 2001 Winter Simulation Conference, pp. 1330–1338. ACM (2001)

20. Ponci, F., Deshmukh, A., Monti, A., Cristaldi, L., Ottoboni, R.: Interface for multi-agent platform systems. In: Proceedings of the IEEE Instrumentation and Measurement Technology Conference, pp. 2226–2230. IEEE (2005)
21. Robinson, C.R., Mendham, P., Clarke, T.: A multiagent approach to manage communication in wis. Journal of Physical Agents 4(3), 489–503 (2010)
22. Sobieh, A., Hou, J.: A simulation framework for sensor networks in j-sim (2003)
23. Titzer, B., Lee, D.K., Palsberg, J.: Avrora: scalable sensor network simulation with precise timing. In: Proceedings of the 4th Int. Symp. on Information Processing in Sensor Networks, pp. 477–482. IEEE (2005)
24. Weyns, D., Schelfthout, K., Holvoet, T., Lefever, T.: Decentralized control of e'gv transportation systems. In: 4th Int. Joint Conf. on Autonomous Agents and Multiagent Systems - Industrial Applications, pp. 67–74. ACM (2005)

Multi-agent Models for Transportation Problems with Different Strategies of Environment Information Propagation

Jarosław Koźlak, Sebastian Pisarski, and Małgorzata Żabińska

AGH University of Science and Technology,
Al. Mickiewicza 30, 30-059 Kraków, Poland
{kozlak,zabinska}@agh.edu.pl, pisarski.sebastian@gmail.com

Abstract. The aim of the work is to show a solution of dynamic transportation problem Pickup and Delivery Problem with Soft Time Windows, taking into account variable travel times between locations. The information about changes is obtained thanks to an algorithm that generates and propagates traffic jams or collects data from the traffic simulator. The model of such a multi-agent system is presented. Different types of propagation of information about changes of travel times for agents constituting a fleet of vehicles for the realisation of transportation requests and different methods of prediction of future actual travel times on road sections are taken into consideration.

Keywords: Transportation Problems, Dynamic Environment, Information Propagation, Prediction.

1 Introduction

A substantial amount of work on agent systems is targeted on building these systems to function properly in the dynamic environment, while adapting themselves to changing situations. The most important problem emerging in such systems is: to gain information about changes as well as accuracy and currentness of the information and quality of the models created to anticipate future changes. Many models of the multi-agent systems were developed considering these aspects (for example [5]), but in our work we are aiming to analyse a transportation problem from this point of view, such a kind of evaluation has not yet been performed according to our knowledge. We are focusing on Pickup and Delivery Problem with Soft Time Windows (PDPSTW) with changing times of travel caused by propagating traffic disturbances. This transport problem belongs to NP-hard problems. PDPTW consists in serving a set of transport requests by a fleet of available vehicles. Particular requests are described by points of pick up and delivery, time windows, when pick up and delivery should occur and required capacity (load). The function of the solution quality is dependent on solution parameters as: number of vehicles, the total distance, the total time

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 145–156, 2013.

of travel, times of waiting (stops) and the penalties, in cases of delay, so the solution of the problem may be considered as multi-criterial. Actions of all agents have influence on the final obtained result. The agents express their cooperative nature by solving imposed tasks and by exploring the environment. It is possible to evaluate the algorithms grading the scale of disturbances. There is also a capability to control the threshold of acceptable level of penalty from the side of each agent, by defining maximal acceptable delays in relation to time or total distance of travel.

Since these solutions may be mapped on different types of problems, e.g. related to tasks scheduling, the obtained conclusions may also be referred to other types of computational problems [4].

In this paper, we extended our solutions worked out so far, with elements related to the propagation of information about changes of environment and implementing appropriate methods of disturbances propagation. In [14] the influence of closure of particular roads on the quality of obtained results was examined, whereas in [8] - the influence of other crisis situations, as well. The problems connected with considering information and reaction on disturbances were examined in [9], where we examined the influence of probabilistic data concerning the spatial distribution of requests on modifications on vehicle routes upon the final solutions gained.

2 Research Domain Overview

Transportation Problems and Heuristic Methods of Their Solving. Transportation problems of the type vehicle routing problem, describing transfer of goods between locations, are widely examined with the use of different heuristic algorithms. In our work we will especially concentrate on the problem with time windows, i.e. Pickup and Delivery Problem with Time Windows (PDPTW), as well as on its special case – Vehicle Routing Problem with Time Windows (VRPTW). For the static PDPTW all requests are known at the moment of the vehicle journey start. The set of test benchmarks was proposed to evaluate the quality of solution of such problems ([13,1]). In the work we particularly concentrate on the dynamic transportation problems, where a request comes during vehicles movement/travel. A review of heuristic solutions of the problem is available in [15].

Multi–agent Models for Solving Transportation Problems. Many multi-agent systems for transport planning have come into existence. One of the best known, and at the same time presenting solutions, which we also refer to in our work, is MARS System [6]. Similarly to meta-heuristic solutions which comprise two phases – constructive heuristics and optimisation heuristics, also in this approach two steps are used: contract-net protocol [16], for initial assigning requests to vehicles and "simulated trading" algorithm [3] for later optimisation of the solution, allowing agent to accept or to get rid of requests which realisation becomes costly for them, chains of such exchanges with the participation of many agents may be constructed in this process.

Perception of Environment Dynamics, Information Propagation and Value. Obtaining and interpreting of information gained from the environment is a substantial domain in examining multi-agent approach and distributed artificial intelligence from the beginning of 1980's. In [12] a concept of the environment consisting of distributed sensor network monitoring of moving vehicles is presented. The problems of application of multi-agent systems for creating distributed networks of sensors was then widely examined, the review of works can be found in [11,17]. Optimal frequency of information exchange, in reference to the policy of performed actions in cooperative multi-agent system was examined in [7].

3 System Model

3.1 Main Model Elements

Our model comprises a dynamic environment and a set of agents of two types. The environment is described by a graph representing a transport network (the chosen essential locations and crossroads, as well as road sections between them) with the current times of travel by particular road sections with information about currently active (reported but not loaded yet) transportation requests, available for loading at a given point. Dynamics of an environment on the one hand - consists in changes of the set of active requests – disappearing loaded requests and the new incoming ones – but on the other hand - in generated changes of times of travel by road sections.

The Dispatcher is responsible for collecting transportation requests and assigning them to Transportation Units, identification of the general situation and adjustment of general working schema to the current situation. Transportation Units representing vehicles, are responsible for request realisation and in some scenarios, also for collecting information about the current environment state. Agents of the level of the model in the system are described by indexes $j\colon j \geq 0 \wedge j \leq n$, where agent $j : j = 0$ is a Dispatcher agent, but agents $j\colon j > 0$ are Transportation Units.

3.2 Model of Dynamic Environment

The environment model, used by the Dispatcher and Transportation Units, includes information about the structure of the transportation network, current, historical and predicted travel times across the road sections, and data about transportation requests, not assigned or picked up, in the system.

Therefore, a dynamic environment in time step t may be described as follows:

$$DynEnv(t) = (\mathbb{G}(t), ReqsA(t), ReqsP(t)) \qquad (1)$$

where:

– $\mathbb{G}(t)$ – total state of the transportation network in the time step (t), contains not only the current state in the time step t – $Gr(t)$, but also information

about the previous states of the network, used for prediction, and future predicted states of network, marked $Gr(k)_{(t)}$, where in the case of historical data $k = t - h_{max}, t - h_{max} + 1, \dots t - 1$, where h_{max} – assumed size of history, and in the case of predicted future changes of network state, $k = t + 1, t + 2, \dots t + p_{max}$ where p_{max} – the maximum range of the predicted steps,
– $ReqsA(t)$ – a set of active requests, which arrived but still are not picked up by transportation units,
– $ReqsP(t)$ – set of requests picked up by vehicles, which still need to be delivered to destination locations,

The given agents j may have only partial and not up to date knowledge, regarding the environment, $DynEnv^j(t)$, in practice it may not include up to date knowledge about travel times across the road sections, incomplete historical information or disturbed prediction results.

Changes of the environment include:

– new request coming: a new request i comes in the time step t: $ReqsA'(t) := ReqsA(t) \cup Reqs_i$, $Reqs_i$ –new request coming in time step t, $ReqsA(t)$ – old set of active requests, $ReqsA'(t)$ – new modified set of active requests,
– picked up request $Reqs_i$ in time step t, a request ceased to be active and is loaded to the vehicle: $ReqsA'(t) := ReqsA(t) \backslash Reqs_i, ReqsP'(t) := ReqsP(t) \cup Reqs_i$
– delivery of request $Reqs_i$ in time step t, a request ceases to be loaded and is not considered anymore in the current state of the system: $ReqsP'(t) := ReqsP(t) \setminus Reqs_i$
– changes of travel time through the road sections: $Gr'(t) = Change(Gr(t))$

3.3 Fundamental Model

Dispatcher Dispatcher (DA). Agent Dispatcher is responsible for sending requests to the vehicles. The representation of dispatcher in time stamp t is expressed by equation 2.

$$DA(t) = (G^0(t), S^0(t), K^0(t), A^0) \tag{2}$$

where:

$G^0(t)$ – a value of goal function for time step t, which depends on the number of vehicles, total travel distance and total penalty for delays,

$S^0(t)$ – state of dispatcher, $S_0 = \{LocAgent(t), ReqsP(t), ReqsA(t)\}$, where $LocAgent(t)$ – set of locations of all transportation unit agents j, defined as follows: $LocAgent_j(t) = \{LocAgent_j(t) \colon j = 1 \dots m\}$, – estimated location of all agents j, m – number of transportation units, $ReqsP(t)$ – set of sets of requests assigned to and picked up to given transportation units j, $ReqsP^{(t)} = \{ReqsP^j(t) \colon j \in [1, m]\}$, $ReqsA(t)$ – set of sets of requests assigned to given transportation units j, but not picked up yet, $ReqsA(t) = \{ReqsA^j(t) \colon j \in [1, m]\}$:

$K^0(t)$ – knowledge of dispatcher, $K^0(t) = \{Env^0(t)\}$, where $Env^0(t)$ - information about transport network in time t, $Env^0(t) = (Gr(t), HGR^{(t)}(LpU))$, $Gr^{(t)}$ – state of transport network in time t and $HGR^{(t)}(LpU)$ – history of previous states of transport network during last LpU changes, $HGR^{(t)}(LpU) = \{Gr^{(k)} : k \geq t - LpU \wedge k \leq t \wedge k \in \mathbf{N}\}$
A^0 - allowed actions of dispatcher.

The allowed actions of dispatcher are as follows:

$$A^0 = \{Allc^0, GtPos^0, SndPos^0, PU^0\} \tag{3}$$

where:

$Allc^0$ – auction of a request and its allocation to a Transportation Unit,

$$Allc^0 : (Rec_i, ReqsV(t), Reqs(t), t) \rightarrow (ReqsV(t) = ReqsV(t) \backslash Rec_i, \\ ReqsA^j(t) = ReqsA^j(t) \cup Rec_i)$$

where j – agent which got a request for realisation, $ReqsV(t)$ – set of not assigned requests, $ReqsA^j(t)$ – set of all requests assigned to given transportation units j
$GtPos^0$ – get info about positions of transportation units, $GtPos^0 : (LocAgent(t)) \rightarrow (LocAgent(t) = \bigcup_{0<j\leq m} Loc^j)$
$SndPos^0$ – send info about agent positions to transportation units,
PU^0 – operation of prediction (described in more detail in section 3.5).

Transportation Unit. Transportation unit TU^j is expressed as follows:

$$TU^j = (G^j(t), P^j(t), K^j(t), S^j(t), A^j), \tag{4}$$

$G^j(t)$ – the goal function, which depends on total travel distance and total penalty,
$P^j(t)$ – a plan represented as a sequence of operations of movements, pickup and delivery operations, acquiring information about change of travel time for given road section,
$K^j(t)$ - knowledge of $j-th$ transportation unit $K^j = \{Reqs^j(t), DynEnv^j(t)\}$ comprising $Reqs^j$ - set of requests, $DynEnv^j(t)$ - information about dynamic environment (see 3.2). environment (transport network).
$S^j(t)$ – state of the j-th transportation unit, $S^j = \{Loc^j, ReqsA^j, ReqsP^j\}$, where Loc^j, – location, $ReqsA^j$ – allocated requests, $ReqsP^j$ – picked-up requests,
A^j - allowed actions of j-th transportation unit.

Allowed actions of the transportation unit j are as follows:

$$A^j = \{Allc^j, Mv^j, Ld^j, Dlv^j, Inf^j, PU^j\} \tag{5}$$

$Allc^j$ - participation in auctions of transport requests
$Allc^j : (ReqsA^j(t), Reqs_i, t) \rightarrow (ReqsA^j(t) = ReqsA^j(t) \cup Reqs_i)$
Mv^j – transfer between locations,
$Mv^j(N_p, N_q) : (Loc^j = N_p) \rightarrow (Loc^j = N_q)$
Ld^j - loading,
$Ld^j : (ReqsA^j(t), ReqsP^j(t), Reqs_i, t) \rightarrow (ReqsP^j(t) = ReqsP^j(t) \cup Reqs_i,$
$ReqsP^j(t) = ReqsP^j(t) \backslash Reqs_i)$
Dlv^j - unloading,
$Dlv^j : (ReqsP^j(t), Reqs_i, t) \rightarrow (ReqsP^j(t) = ReqsP^j(t) \backslash Reqs_i)$
Inf^j : – gather information about travel times for current road section, in time t (q – current edge/road section),

$Inf^j : (T_{E_q^j(t)}, q, t) \rightarrow Env^j = Env^j \cup (T_{E_q^j(t)})$
PU^j – operation of prediction, which produces the predicted values of travel times for road sections for next p_{max} steps, $PU^j : (Env^j) \rightarrow (Env^j)'$

3.4 Model of Information about Propagation of Environment Changes

In the realised environment, there are several methods of propagation of information about disturbances taking place or being noticed, this information reaches Dispatcher and Transportation Units. The model of the information propagation about dynamic environment is expressed as follows $DynEnvInfo(t) = (DynEnv(t), p)$, where $DynEnv(t)$ is the model of dynamic environment described in section 3.2 and p is an assumed mode of information propagation, which determines the method, how agents become acquainted with changes of travel times associated with given road sections.

The following modes p of information propagation are currently considered:

- *immediately* – (i) – dispatcher and all transportation units obtain globally immediate information about every change of travel times in any of the road sections,
- *after noticing* – (an) the information about change is disseminated to all agents only when a transportation unit discovers it (it means drives the road section with changed travel times),
- *after time* – (pX) – changes are disseminated to all agents after the given time period X, if a transportation unit notices the change earlier, it takes it into consideration in its plan, but does not inform other agents.

3.5 Models of Prediction of Environment Changes

The Dispatcher and Transportation Units may either not predict the future state of the environment, considering current and historical information, or predict them using one of several prediction methods. If prediction algorithms are used, the considered size of historical changes may influence the results. Predicting agents are equipped with a prediction unit PU which is represented as follows: $PU = PU(L_{PU}, A_{PU})$, where

- L_{PU} – number of remembered last changes of the road graph,
- A_{PU} – a prediction algorithms, the following solutions are implemented: *Standard* (lack of prediction), *Average* (basic mean value), *MovingAverage* (exponential moving average).

4 Realisation

The main elements of the system are presented in fig. 1.

Simulation control (Test Agent) – a component responsible for reading input files and generating the output ones. Its task is a simulation initialisation and sending simulation progress time markers. It also performs a function of the "outer world" from the point of view of other agents, it transfers incoming requests, as well as takes care of propagation of changes in the road network graph.

Data about agents in the system (InfoAgent) – the agent, which deals with creation (with the use of JADE [2] platform tools) of component agents of Transportation Units. It also stores identifiers of created agents.

Distribution of requests (Dispatcher) – it receives incoming requests from the control unit. Then they are queued in the optimal sequence (different policies) in such a way to increase probability of optimal routes creation after assigning requests to Transportation Units. Each request is sent to the existing Transportation Units with a query about costs of realisation. The Unit offering the lowest cost is chosen. When the request cannot be realised using the current set of vehicles, it initiates the process of negotiation (level 2 optimisation process), based on our implementation of a Simulated Trading [3] algorithm. It gathers schedules of all Transportation Units, modifies them, and sends them back - updated. If it is still not possible to serve the request, it orders the creation of a new Transportation Unit.

Transportation Units – they serve requests assigned by the Dispatcher. After receiving the new request they calculate the cost of its realisation and send it back to the Dispatcher. Additionally Transportation Unit transfers information on its state to the simulation control module, and to the Dispatcher. Apart from this, it has also the capability to change its route construction strategy during the course of simulation. After allocating new requests, the Transportation Units start an optimisation process (level 1), they offer a given number of requests allocated to them, that mostly increase the cost of their route, to other Transportation Units. If there is another agent which can serve a request with lower costs than its current owner, the request is moved to its route.

In the system, two types of disturbance generation were realised.

Simple generator of disturbances. It is based on the concept of virtual vehicles, which move along graph edges, carrying disturbances by themselves. The disturbance generator functions in the following way:

- initialisation – parameters of each vehicle are drawn from available values set during the initialisation of the generator. They are: departure time, worsening of traffic factor, the path to be travelled by the given vehicle,

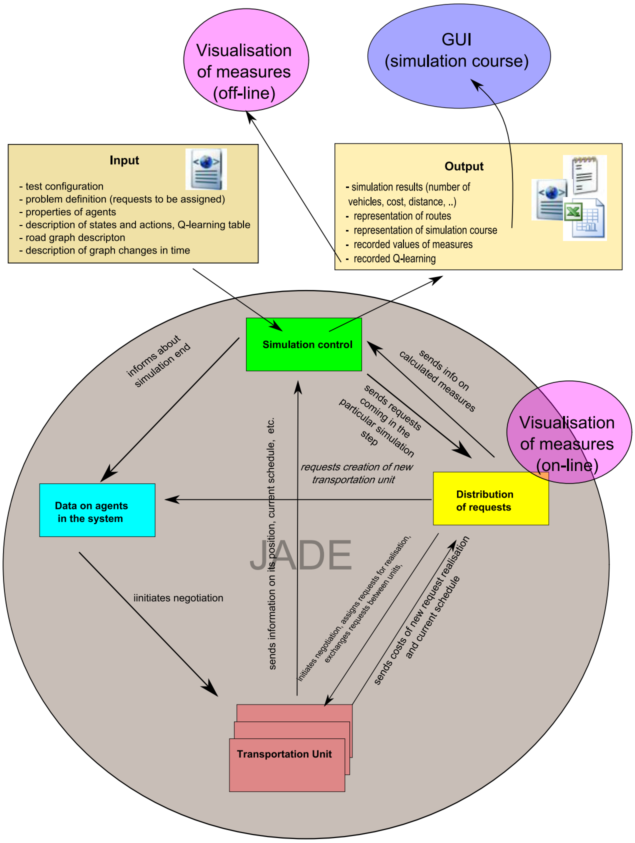


Fig. 1. General architecture of the system

- vehicles are moved in the loop until the deadline iterations, when the vehicle enters the given road section, it modifies the time of travel using its worsening factor, the change remains until the vehicle leaves the road section,
- the state of the graph with current travel times through the road sections is recorded every given number of time steps.

Use of Configuration Data Generated by Multi-agent Traffic Simulator. The second method of gathering information on the changes of traffic states and changes of travel times through road sections are obtained from the multi–agent traffic simulator, described in [10]. It uses extended multi-lane Nagel-Schreckenberg traffic model, and adaptive algorithm for traffic lights on intersections.

5 Results

Experiments were carried out for different types of graphs as well as different grades of disturbances. For such defined conditions tests were carried out with the use of different types of information on propagation and different prediction algorithms. One group of tests (*tests1*) was performed with disturbances generated by virtual vehicles, the second for travel times generated using traffic modelling system (*tests2*).

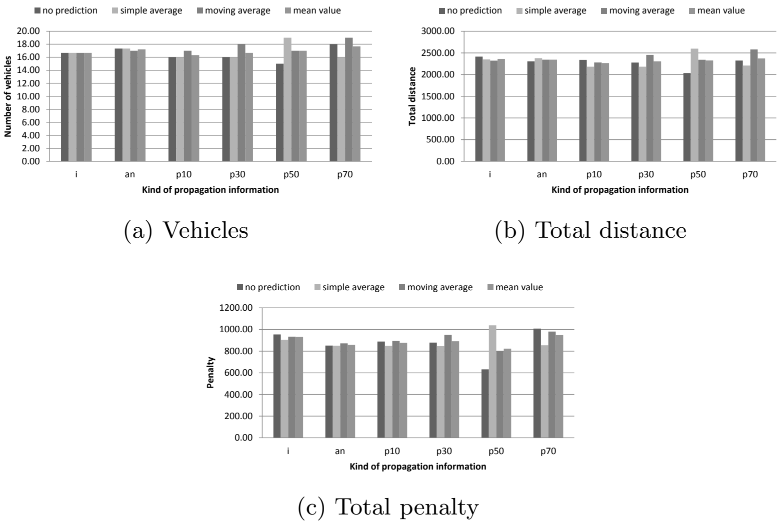
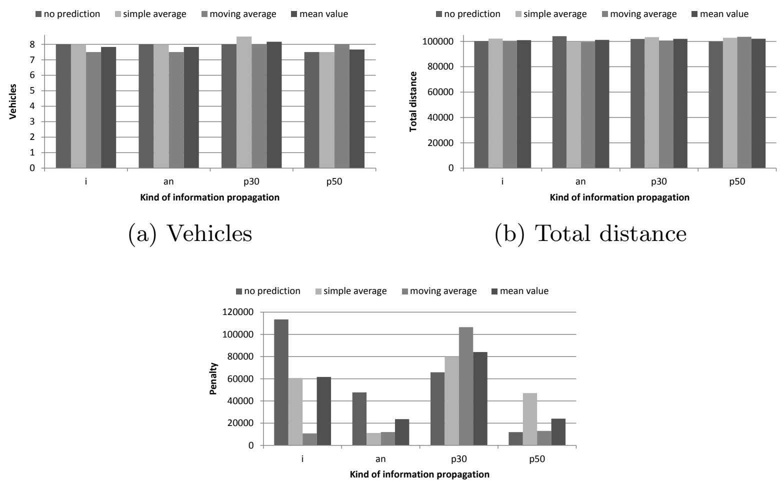


Fig. 2. Results for high disturbances. Different methods of propagation information about changes and different prediction algorithms.

Table 1. Average values for different strength of traffic disturbances

disturbance strength	vehicles	distance	penalty	non-performed requests
veh100	16.94	2351.95	888.19	0.00
veh200	19.28	2770.52	1014.67	0.00
veh300	19.17	2732.52	892.58	3.00
veh400	17.25	2499.25	1034.25	7.58
veh500	18.58	2354.41	974.83	11.00

For *tests1* average values of vehicle numbers were shown, as well as the total distance and penalties gained for several simulation runs, for each considered methods of information propagation and predictions (fig. 2).



(a) Vehicles (b) Total distance

(c) Total penalty

Fig. 3. Results obtained for different information propagation and different prediction algorithms using data from road traffic simulation

The following methods of information propagation about changes of travel times were used: immediate (*i*), after noticing by vehicle which entered the given edge (*an*), or with delay of 10, 30, 50 and 70 time units (*pXX*). For each method of information propagation the results are given for the following situations: *no prediction*, prediction using the following *simple average* and *moving average* methods, and *mean value* obtained using all methods.

One can see that an average number of used vehicles usually slightly increases with the decrease of information accuracy. One of the prediction methods, especially *simple average* in most cases enables obtaining better results than in case without prediction. Thus we can say, that although the results are to some extent burdened by random generation of disturbances, usually more accurate information slightly favours better solutions.

Analysing obtained results from the point of view of two criteria viz. (i) solution quality or vehicles number, then the distance and (ii) penalty from delays, one can distinguish non-dominated solutions making Pareto front. They are: simple average and moving average for immediate information flow, simple average for information obtained after 10, 30 and 70 time units, no prediction for information obtained after 50 time units, all three predicting methods for the information obtained after the vehicle visits of the edge. So, the use of methods of predicting changes, even despite their simplicity (lack of consideration e.g. identification of patterns recognised in history) usually allows us to obtain better solutions than in cases without prediction.

The important question is also how the increase of the level of the traffic disturbances influence the average results obtained by agents (tab. 1). We considered the disturbances generated by 100, 200, 300, 400 and 500 virtual vehicles. For each level of disturbances average values were calculated, considering the problems with all kinds of propagation of information about changes and three considered prediction strategies. For disturbances generated by 100 and 200 vehicles all requests are performed, for 200 vehicles the number of vehicles and total distance increase. For cases with 300, 400 and 500 vehicles appear and increase requests which are not possible to be served in a given time horizon.

For the *tests 2* one can notice that differences for different methods of information propagation are small. It is caused by more realistic and steady changes of the traffic characteristic. Considering the non-dominated solutions for each kind of information propagation, it is a moving average for immediate information propagation, moving and simple average for information obtained after vehicle visits the edge, moving average and no prediction for information obtained after 30 time units and no prediction for obtained after 50 time units. It is worth noting that for the information obtained after much longer time period – 100 time units – the number of vehicles for no prediction increased to 9, while for the both prediction methods it remained at the level of 8 vehicles. Also the total distance for no prediction increased to 112 thousands, remaining at about 100 thousands and 103 thousands for simple and moving average respectively. The penalty for no prediction case increased to a very large value, more than 300 thousands, while for the prediction cases, it remained at the usual low level.

6 Conclusions

Within frames of our works the model and environment to solve dynamic transport problems with changes of travel times were developed and different methods of propagation of information about changes and travel time prediction, for different strengths of disturbances were evaluated.

Analysing the results for more natural evolution of traffic conditions (*test2*) one can notice, that for the majority of cases the impact of the choice of prediction methods and information propagation methods was weak, but for very accurate and very delayed information results obtained without prediction were significantly worse. It seems that for up-to-date information about traffic conditions, prediction may help to obtain better results allowing to fit the plans better to the future real situation. For very inaccurate information, prediction often allowed reduction of differences between the real traffic conditions and the conditions considered during planning, while for the no prediction technique this gap remained very large.

For stronger changes of traffic conditions (*test1*), the results (number of vehicles and distance) are usually better for cases with prediction, especially when the information is up-to-date (i, p10, p30). When the level of disturbances exceeds a given threshold (for the tests it was 300 virtual vehicles causing increases in travel times) it is no longer possible to handle all requests, the number of non-served requests increases together with the further increase of disturbances.

Further work will comprise extension of the set of available optimisation algorithms and their configuration. Another direction of work will be applying more complex regression methods for predicting traffic conditions, such as clustering and regression trees techniques.

References

1. Benchmarks - Vehicle Routing and Travelling Salesperson Problems, http://www.sintef.no/Projectweb/TOP/Problems/PDPTW/Li--Lim-benchmark/
2. Java Agent DEvelopment Framework, http://jade.tilab.com/
3. Bachem, A., Hochstättler, W., Malich, M.: The simulated trading heuristic for solving vehicle routing problems. Discrete Appl. Math. 65, 47–72 (1996)
4. Beck, J.C., Prosser, P., Selensky, E.: Vehicle routing and job shop scheduling: What's the difference? In: Proc. of the 13th International Conference on Automated Planning and Scheduling, pp. 267–276 (2003)
5. Cetnarowicz, K., Gruer, P., Hilaire, V., Koukam, A.: A formal specification of M-agent architecture. In: Dunin-Keplicz, B., Nawarecki, E. (eds.) CEEMAS 2001. LNCS (LNAI), vol. 2296, pp. 62–72. Springer, Heidelberg (2002)
6. Fischer, K., Muller, J., Pischel, M.: Cooperative Transportation Scheduling: an Application Domain for DAI. Applied Artificial Intelligence, 1–33 (1996)
7. Goldman, C.V., Zilberstein, S.: Optimizing information exchange in cooperative multi-agent systems. In: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2003, pp. 137–144. ACM, New York (2003)
8. Konieczny, M., Koźlak, J., Żabińska, M.: Multi-agent crisis management in transport domain. In: Allen, G., Nabrzyski, J., Seidel, E., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2009, Part II. LNCS, vol. 5545, pp. 855–864. Springer, Heidelberg (2009)
9. Koźlak, J., Créput, J.C., Hilaire, V., Koukam, A.: Multi-agent approach to dynamic pick-up and delivery problem with uncertain knowledge about future transport demands. Fundamenta Informaticae: Annales Societatis Mathematicae Polonae 71(1) (2006)
10. Koźlak, J., Dobrowolski, G., Kisiel-Dorohinicki, M., Nawarecki, E.: Anti-crisis management of city traffic using agent-based approach. Journal of Universal Computer Science 14(14), 2359–2380 (2008)
11. Lesser, V., Tambe, M., Ortiz, C.L. (eds.): Distributed Sensor Networks: A Multiagent Perspective. Kluwer Academic Publishers, Norwell (2003)
12. Lesser, V., Corkill, D.: The Distributed Vehicle Monitoring Testbed: A Tool for Investigating Distributed Problem Solving Networks. AI Magazine 4(3), 15–33 (1983)
13. Li, H., Lim, A.: A Metaheuristic for the Pickup and Delivery Problem with Time Windows. In: Proc. of 13th IEEE Int. Conf. on Tools with Artificial Intelligence, ICTAI 2001 (2001)
14. Nawarecki, E., Koźlak, J., Dobrowolski, G., Kisiel-Dorohinicki, M.: Discovery of crises via agent-based simulation of a transportation system. In: Pěchouček, M., Petta, P., Varga, L.Z. (eds.) CEEMAS 2005. LNCS (LNAI), vol. 3690, pp. 132–141. Springer, Heidelberg (2005)
15. Pillac, V., Gendreau, M., Guérat, C., Medaglia, A.L.: A review of dynamic vehicle routing problems. Tech. Rep. CIRRELT-2011-62, CIRRELT (October 2011)
16. Smith, R.G.: The contract net protocol: high-level communication and control in a distributed problem solver. IEEE Transactions on Computer, 1104–1113 (December 1980)
17. Vinyals, M., Rodriguez-Aguilar, J.A., Cerquides, J.: A survey on sensor networks from a multiagent perspective. Comput. J. 54(3), 455–470 (2011)

How to Build the Best Macroscopic Description of Your Multi-Agent System?

Robin Lamarche-Perrin[1,2], Yves Demazeau[1,3], and Jean-Marc Vincent[1,4]

[1] Laboratoire d'Informatique de Grenoble, France
[2] Université de Grenoble
[3] CNRS
[4] Université Joseph Fourier
Robin.Lamarche-Perrin@imag.fr, Yves.Demazeau@imag.fr,
Jean-Marc.Vincent@imag.fr

Abstract. The design and debugging of large-scale MAS require abstraction tools in order to work at a macroscopic level of description. Agent aggregation provides such abstractions by reducing the complexity of the microscopic description. Since it leads to an information loss, such a key process may be extremely harmful for the analysis if poorly executed. This paper presents measures inherited from information theory to evaluate abstractions and provide the experts with feedback regarding the quality of generated descriptions. Several evaluation techniques are applied to the spatial aggregation of an agent-based model of international relations. The information from on-line newspapers constitutes a complex microscopic description of agent states. Our approach is able to evaluate geographical abstractions used by the domain experts in order to provide efficient and meaningful macroscopic descriptions of the world global state.

Keywords: Large-scale multi-agent systems, agent aggregation, macroscopic description, information theory, geographical and news analysis.

1 Introduction

Because of their increasing size, complexity and concurrency, current multi-agent systems (MAS) can no longer be understood from a microscopic point of view. Design, debugging and optimization of such large-scale distributed applications need tools that proceed at a higher level, with insightful abstractions regarding the global system dynamics. Among abstraction techniques (dimension reduction, subsetting, segmentation, clustering, and so on [1]), this paper focuses on *data aggregation*. It consists in losing some information about the agent level to build simpler yet meaningful macroscopic descriptions. Such a process is not trivial for the data interpretation. In particular, unsound aggregations may lead to a critical misrepresentation of the MAS behavior. Hence, we have to determine what are the *good* abstractions and how to properly use them. At each stage of MAS development, aggregation processes should be carefully monitored and feedback should be provided regarding the quality of generated macroscopic descriptions.

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 157–169, 2013.

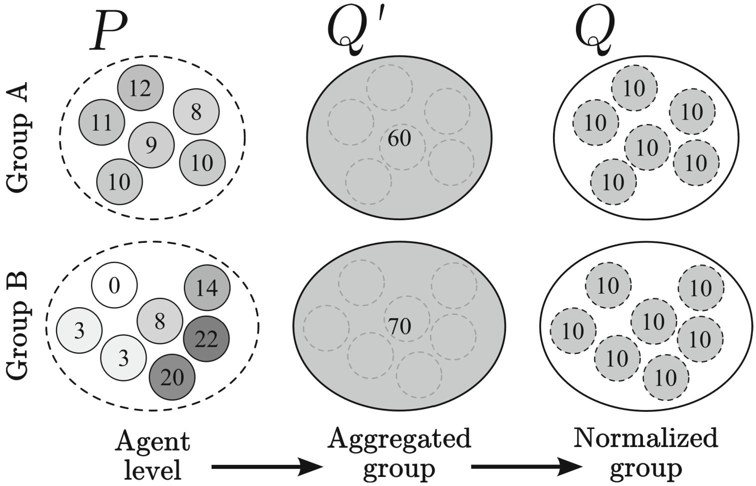


Fig. 1. Averaging the behavior of groups of agents may reduce the redundant information (group A) or it may lead to an unwanted information loss (group B)

A simple example can demonstrate how critical an aggregation can be. Fig. 1 shows two groups of agents that may be simplified by two abstract entities with an average behavior. Intuitively, group A constitutes a *good* abstraction since the induced global behavior is relatively similar to the microscopic one, unlike group B. Hence, aggregation of redundant information should be encouraged to reduce the description complexity (group A), but details regarding heterogeneous behaviors should be preserved in order to control the information loss (group B).

Very little work has been done in the MAS community to quantify such aggregation properties. The main contribution of this paper consists in introducing measures from information theory (Kullback-Leibler (KL) divergence [2] and Shannon entropy [3]) to clarify the notion of *good* aggregation. From these measures, we provide generic feedback techniques and an algorithm that builds multi-resolution descriptions out of hierarchically organized MAS. These techniques and algorithms are applied to the agent-based modeling of international relations: agents represent countries, and their behavior is extracted from online newspapers. Geographers exploit multi-level aggregates to build statistics regarding world areas. We show how these geographical abstractions should be used to better understand the system states and, with further research, its dynamics. This ambitious GEOMEDIA project is conducted in collaboration with experts from the CIST (*Collège International des Sciences du Territoire*, Paris).

Section 2 presents the work related to the main concern of this article. Section 3 presents the agent-based model of the GEOMEDIA application. Section 4 introduces KL divergence to estimate *information loss* and section 5 Shannon entropy to estimate *complexity reduction*. Section 6 shows how these measures can be combined to identify *best* aggregations and to build multi-resolution representations. Section 7 concludes this paper and gives some perspectives.

2 Related Work

Aggregation can take place in every stage of a MAS development: from its design to its use. Even if abstraction techniques may differ, each stage should carefully take into consideration the quality of the aggregations. First, from a software perspective, this section shows that very few research efforts have been done to tackle this issue. (1) Most classical simulation platforms and monitoring systems do not even provide the user with abstraction tools; (2) some do handle the issue, but are still at an early stage of thought. Secondly, on a theoretical aspect, this section explains why classical techniques (*e.g.* data clustering, graph analysis) are not entirely satisfying to build consistent abstractions. In this regard, our approach should rather be compared to recent work in multi-level MAS [4] to which it may provide a formal and quantitative framework.

In a comprehensive survey of agent-based simulation platforms [5], Railsback *et al.* evaluate some of them by testing classical features of MAS modeling and analysis. Unfortunately, the abstraction problem is not tackled by this survey, thus indicating that such considerations are seldom if ever taken into account. Most platforms (Java Swarm, Repast, MASON, NetLogo and Objective-C Swarm) are limited to the microscopic simulation of agents. Railsback warns about the lack of "a complete tool for statistical output" in these platforms [5]. The provision of global views on the MAS macroscopic behavior thus constitutes an on-going research topic. Some tools for large-scale MAS monitoring address this issue by using aggregated data or visual abstractions to reduce the complexity of execution traces [6,7]. However, these abstractions are either limited to the simplification of agents internal behavior, and do not tackle multi-agent organizational patterns, or they do not provide feedback regarding the quality of such abstractions.

Some techniques from graph analysis and data clustering build groups of agents based on their microscopic properties [8,9,10]. Such considerations may meet ours from a theoretical point of view, but the approach presented in this paper supports a very different philosophy: *abstractions should be consistent with the macroscopic semantics of the system.* We claim that, to be meaningful, the aggregation process needs to rely on high-level concepts provided by the domain experts. Hence, our approach should rather be compared with research on multi-level agent-based models [4]. These works openly tackle the abstraction problem by designing MAS on several levels of organization according to expert definitions. Such approaches aim at reducing the computational cost of simulations by reducing the amount of detail. The measures and techniques presented in this paper may provide a formal and quantitative framework to support such a research effort.

To conclude, aggregation techniques should be more systematically implemented on MAS platforms in order to handle large-scale systems. They should combine consistent macroscopic semantics from the experts and feedback regarding the abstractions quality. For example, in this paper, abstractions used by geographers are evaluated according to their information content.

3 Agent-Based Modeling of International Relations

This section presents the GEOMEDIA agent-based model. It consists in the microscopic description of countries with agents and the macroscopic description of world areas with groups and organizations.

3.1 Microscopic Data: The Agent Level

Let A be a set of agents. It constitutes the MAS microscopic level. Visualization tools aim at displaying and explaining *variables* regarding these agents: their behavior and internal states, the events they are associated with, the messages they exchange, and so on. Given a variable v, the set of values $\{v(a)\}_{a \in A}$ forms the *microscopic description* of the system (illustrated by distribution P in Fig. 1).

In the GEOMEDIA project, we are interested in the analysis of world international relations. In that context, we make the assumption that citations or co-citations of countries, within news, are good indicators to represent and understand their relations. For example, we may assume that an often-cited country is likely to politically interact with the newspaper country. In our model, the microscopic level of agents is constituted of 168 countries. Information regarding their behavior has been extracted from 70 RSS feeds of English-language newspapers, from May 2011 to September 2012. The experiments in this paper focus on a very basic variable, `citations_nb`: the number of articles that name a country, and three newspapers: the Vancouver Sun (`feed_CAN`), the Daily Mail (`feed_GBR`), and the Philippine Daily Inquirer (`feed_PHL`).

3.2 Macroscopic Data: Groups and Organizations

A *group* $G \subset A$ is subset of agents that are members of a consistent organizational pattern. It can be interpreted as an *abstract agent* that sums up the behavior of its underlying agents. Hence, groups satisfy a recursive definition: a group is either an agent or a set of groups. Variables are defined on groups according to an aggregation operator: sum, mean, median, extrema, and so on [1]. In our case, since we work with *extensive* variables (*i.e.* variables that are proportional to the aggregate size), $v(G)$ is the *sum* of the values of the underlying agents: $v(G) = \sum_{a \in G} v(a)$ (see Q' in Fig. 1).

We define an *organization* O as a set of groups that constitutes a *partition* of the agent set A. Thus, in the scope of this paper, each agent is always a member of one and only one group. The set of group values $\{v(G)\}_{G \in O}$ composes a *macroscopic description* of the system wrt an organization. It simplifies the variable distribution, from the detailed microscopic description (P in Fig. 1) to an aggregated one (Q'). When comparing both descriptions, an assumption is made regarding the underlying distribution of the aggregated values (*e.g.* uniform, geometric or Gaussian distribution). In our case, we consider that each agent has the same weight within the aggregate. It is thus underlined that aggregated values are *uniformly distributed* over the agents (from Q' to Q). Consequently, as illustrated in Fig. 1, some groups are more suitable than others for the analysis. For example, using group A seems relevant since P is close to Q, unlike group B. Hence, organizations should be carefully chosen to provide accurate high-level abstractions. In particular, they should only aggregate homogeneous and redundant distributions. The next section presents a measure to quantify such a property.

Groups and organizations can be derived from semantical aspects of the agent space. In a geographical context, social, political, and economic organizations of the world are often used. However, in this paper, we focus on *topological* organizations, in order to be consistent with geographical maps of the world. Groups thus aggregate nearby territories. In the following experiments, we consider two hierarchical organizations of world countries, namely `WUTS` [11] and `UNEP` [12]. They define multi-level nested groups used by geographers to build global statistics about world areas, from the microscopic level of agents to the full aggregation (see [11] for a detailed presentation of these multi-scale organizations).

4 KL Divergence as a Measure of Organization Quality

Among classical similarity measures, Kullback-Leibler (KL) divergence [2] is of high interest because of its interpretation in terms of information content. This section shows how it can be exploited to provide feedback regarding the quality of groups and organizations.

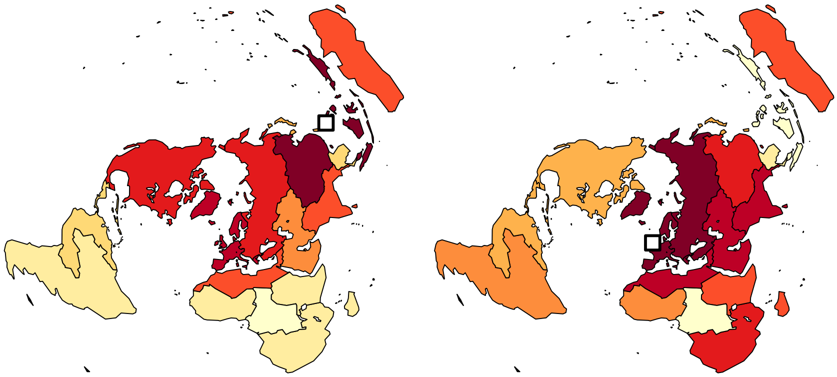
4.1 Kullback-Leibler Divergence

KL divergence measures the number of bits of information that one loses by using an approximated distribution Q to encode the citations of countries, instead of using the detailed source distribution P. In other words, KL divergence estimates the information that is lost by the aggregation process. As we assume that aggregated values are uniformly distributed among underlying agents, a group whose internal distribution is very homogeneous (group A) will have a low divergence (*i.e.* a low information loss), and conversely (group B).

From the KL formula [2], we define *divergence* (or information *loss*) of a group G as follows (more details can be found in [13]):

$$\text{loss}(G) = \sum_{a \in G} \frac{v(a)}{v(A)} \times \log_2 \left(\frac{v(a)}{v(G)} \times |G| \right) \tag{1}$$

where $|G|$ is the size of the group (*i.e.* the number of aggregated agents), $v(G)$ is the sum of the aggregated values and $v(A)$ is the sum of all values (*i.e.* the total number of citations). KL divergence is expressed in bits/citation (or b/c). It verifies the *sum property* [14], meaning that the divergence of disjoint groups is the sum of their divergences. Therefore, for an organization O, we have: $\text{loss}(O) = \sum_{G \in O} \text{loss}(G)$.



(a) `feed_PHL` (Philippines) (b) `feed_GBR` (UK)

Fig. 2. Spatial variations of KL divergence for groups of the `WUTS_3` organization (the darker, the higher). Newspaper locations are indicated by white squares.

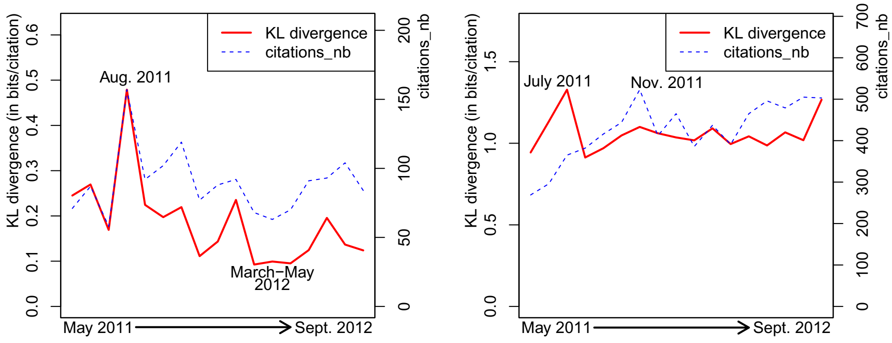
4.2 Groups Quality Is Correlated with the Source of Information

This first experiment aims at showing an essential feature of abstractions: their quality depends on the context of the analysis. Fig. 2 presents the KL divergence of groups from the `WUTS_3` mesoscopic organization, for two newspapers. The darker a group is, the less homogeneous its internal distribution is.

For the investigated dataset, we notice that groups *in which* newspapers are located have high information loss, as for groups that are located *close to* the newspaper (*e.g.* the `Eastern Asia` group in Fig. 2(a)) or that contain agents that are culturally or politically *related to* the newspaper country (*e.g.* `Southern Africa` in Fig. 2(b)). This can be explained by the fact that, for a newspaper, close or related agents may have very divergent behaviors, whereas distant agents are more or less the same. We do not aim at proving that such an hypothesis is universally verified, but at showing that groups should be chosen with respect to the dataset. In our case, this is partly correlated with the source of information. As a consequence, if an analyst uses distributed probes to observe a MAS, she does not want to use only a single abstraction pattern to summarize the information. This is consistent with the *subjectivist* account of emergence, according to which emergent phenomena strongly rely on the observation process [15].

4.3 Groups Quality Varies with Time

Fig. 3 presents the variation of KL divergence and `citations_nb` for two groups of countries on a monthly basis. Fig. 3(a) shows that a group can have a poor quality on specific time periods (*e.g.* August 2011) and high quality on others (*e.g.* from March to May 2012). Abstractions should then be chosen wrt the analyzed time period. Fig. 3(b) shows that the divergence variation is not strictly correlated to the `citations_nb` variation (*e.g.* July 2011 and Nov. 2011). Henceforth, citations number may not be a sufficient criterion for group evaluation.



(a) W. Asia & N. Africa group (b) Europe & N. Asia group

Fig. 3. Time variation of the KL Divergence and the citations number for two groups of agents (for feed_GBR)

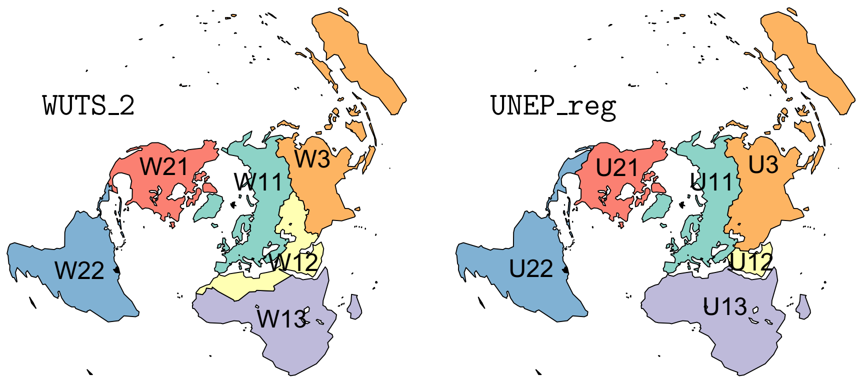


Fig. 4. Two organizations of the agents space in six similar (but not equivalent) groups: locations of the N. African agents, the W. Asian agents and the Mexico agent differ

4.4 Comparing Two Similar Organizations

The purpose of this third experiment is to compare two mesoscopic agent organizations: WUTS_2 and UNEP_reg (see Fig. 4). First, a global comparison can decide which organization is the *best* according to KL divergence. The induced information loss is compared with the total quantity of information contained in the microscopic description, given by the Shannon entropy [3], to give the percentage of lost information.

	feed_CAN	feed_GBR	feed_PHL
WUTS_2	1.80 b/c (62.1%)	1.46 b/c (26.4%)	2.07 b/c (51.0%)
UNEP_reg	1.57 b/c (54.1%)	1.51 b/c (27.3%)	2.26 b/c (55.7%)

It appears that, both for feed_GBR and feed_PHL, divergence is slightly lower for WUTS_2 than for UNEP_reg. Hence, if one should choose between these two organizations, WUTS_2 should be preferred. However, for feed_CAN, UNEP_reg is better. Once again, abstractions should be chosen wrt the source of information.

One can perform a more subtle analysis in order to determine the groups *best* shapes. For example, we notice in Fig. 4 that U22 = W22 ∪ Mexico and W21 = U21 ∪ Mexico. Hence, one may ask: what is the best location of the Mexico agent? Should it be aggregated with the Northern America group (W21/U21) or with the Latin America one (W22/U22)? For feed_GBR, we have:

$$\text{loss}(\texttt{W21}) + \text{loss}(\texttt{W22}) = 0.048 \text{ b/c } < 0.055 \text{ b/c } = \text{loss}(\texttt{U21}) + \text{loss}(\texttt{U22})$$

Thus, the citations number of the Mexico agent is closer to those of the Northern America agents. Mexico should be grouped accordingly. This technique allows to evaluate and choose the shape of abstractions used by the experts.

5 Complexity Reduction of Organizations

The information content cannot be increased by the aggregation process. Hence, for any pair of disjoint groups, we have: $\text{loss}(G_1 \cup G_2) \geq \text{loss}(G_1) + \text{loss}(G_2)$. This means that, if we only rely on KL divergence, the more detailed is always the better. Hence, we need a measure that also expresses what one *gains* with the aggregation. To do so, this section presents two measures of *complexity reduction*. They estimate the information quantity that one saves by encoding a group G rather than its underlying agents: $\text{gain}(G) = \left(\sum_{a \in G} Q(a)\right) - Q(G)$, where Q estimates the quantity of information needed to represent an agent or a group.

5.1 Number of Encoded Values

One way of measuring information quantities consists in estimating the number of bits needed to encode the values of a given description. We suppose that it is constant for each agent or group: $Q(a) = Q(G) = q$, where q depends on the data type of the encoded values. Hence, for a group, we have: $\text{gain}(G) = (|G| - 1) \times q$. It is a basic complexity measure, but it fits well with classical visualizations (as for the maps of this paper) since the number of displayed values defines the granularity of the visualization.

5.2 Shannon Entropy

The number of encoded values *only* depends on the groups partitioning. In contrast, Shannon entropy *also* depends on the variable distribution. It is a classical complexity measure that is consistent with KL divergence: it can be defined as *the divergence from the uniform distribution* [2]. Briefly, entropy evaluates the information quantity needed to encode *each citation* (and not only the citations number for *each agent*). Based on Shannon's formula [3], we define the *entropy reduction* (or *gain*, in bits/citation) of a group G as follows:

$$\text{gain}(G) = \left(\frac{v(G)}{v(A)} \log_2\left(\frac{v(G)}{v(A)}\right)\right) - \sum_{a \in G}\left(\frac{v(a)}{v(A)} \log_2\left(\frac{v(a)}{v(A)}\right)\right) \quad (2)$$

The choice of either one of these complexity measures depends on the performed analysis. *Shannon entropy* should rather be used for the visualization of individuated citations, whereas *the number of values* is more consistent with the visualization of aggregated values. In any case, techniques presented in this paper are meant to be generic. They can be used with any complexity measure as long as it fits with some algebraic properties (see [13] for more details).

6 Multi-resolution Organizations of MAS

As a conclusion to the previous sections, finding a *good* organization relies on two aspects: the *gain* and the *loss* induced by the aggregation of agents into an average behavior. Choosing an organization thus consists in finding a compromise between a complexity reduction and an information loss.

6.1 Parametrized Information Criterion

A *parametrized Information Criterion* can express the trade-off between complexity reduction and information loss for a given group G:

$$\mathrm{pIC}(G) = p \times \mathrm{gain}(G) - (1 - p) \times \mathrm{loss}(G) \tag{3}$$

where $p \in [0, 1]$ is a parameter used to balance the trade-off. For $p = 0$, maximizing the pIC is equivalent to minimizing the loss: the user wants to be as precise as possible (microscopic level). For $p = 1$, she wants to be as simple as possible (full aggregation). When p varies from 0 to 1, a whole class of nested organizations arises. The analyst has to choose the ones that fulfill her requirements: between the expected amount of details and the computational resources available for the analysis.

Fig. 5 presents such a two-dimensional evaluation of the groups of the `WUTS_3` organization. By comparing KL divergence and entropy reduction, one can easily

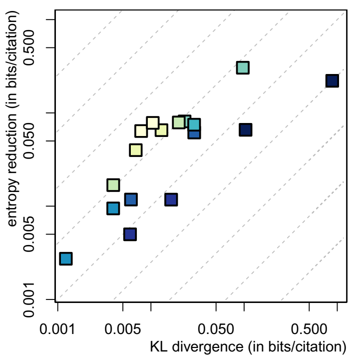


Fig. 5. Comparison of KL divergence and entropy reduction (on logarithmic scales) for groups of `WUTS_3` (for `feed_PHL`)

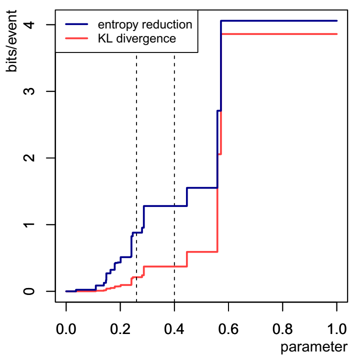


Fig. 6. Variation of the KL divergence and the entropy reduction of best organizations as p varies from 0 to 1

spot groups that have a good gain/loss ratio. The closer a group is to the top-left corner (light squares), the more its complexity reduction compensates its information loss, whereas bottom-right groups have a poor gain/loss ratio and should not be aggregated (dark squares).

6.2 Organizations within a Hierarchy

Given a value of p, *best* organizations are those that maximize the information criterion. Clustering techniques, using *gain* and *loss* measures as distances, could find such optimal partitions. However, results may have very little meaning for the MAS analysis since agents would be aggregated regardless of their location within the system. In contrast, we assume that, in most spatial MAS, there is a correlation between topology and behavior. Hence, we propose that organizations should fit with topological constraints. In other agent-based applications, such constraints can be derived from *semantic* properties of the system (and not necessarily *topological* properties).

In this subsection, we consider hierarchically organized MAS. A *hierarchy* H is a set of nested groups, defined from the microscopic level (each agent is a group) to the whole MAS (only one group). The number of possible multi-resolution organizations within such a hierarchy *exponentially* depends on the number of levels. For `UNEP` (3 levels) and `WUTS` (5 levels), we respectively have 1.3×10^6 and 3.8×10^{12} possible organizations. Finding the best one can thus be computationally expensive. Algorithm 1 below finds topologically-consistent organizations that maximize our information criterion. Its complexity *linearly* depends on the number of groups in the hierarchy (respectively 196 and 231 groups) by doing a classical linear search within the branches of the hierarchy. Indeed, according to the *sum property* [14] of our information-theoretic measures (see subsection 4.1), each branch can be independently evaluated.

This algorithm has been executed on the `WUTS` hierarchy for the `feed_PHL` newspaper. As we increase the gain/loss parameter p, complexity decreases and divergence increases (see Fig. 6). For $p = 0$, all agents are displayed (see Fig. 7).

Algorithm 1. linearly finds best organizations within a hierarchy

Require: A hierarchy H and a trade-off parameter p in $[0, 1]$.
Ensure: An organization made of groups in H that maximizes the pIC.

function FINDBESTORGANIZATION(H, p)
    if H contains only one group G **then return** $\{G\}$
    $G \leftarrow$ biggest group of H
    bestOrganization $\leftarrow \emptyset$
    for each direct subhierarchy S of H **do**
        $aux \leftarrow$ FINDBESTORGANIZATION(S, p)
        $bestOrganization \leftarrow$ UNION($bestOrganization, aux$)
    if pIC of $\{G\} > pIC$ of $bestOrganization$ **then return** $\{G\}$
    else return $bestOrganization$
end function

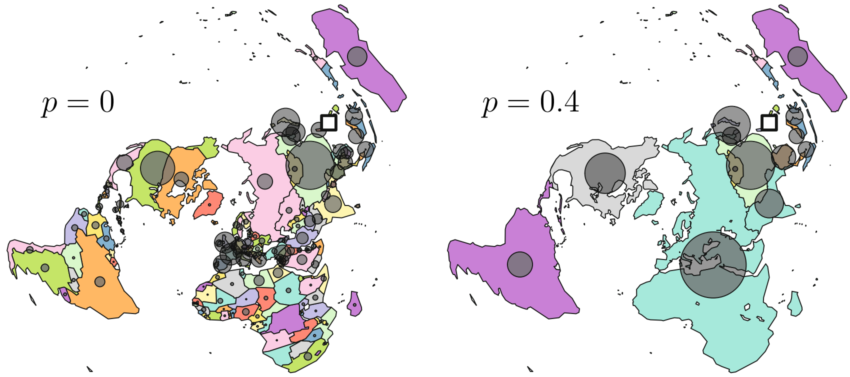


Fig. 7. Two multi-resolution organizations within the `WUTS` hierarchy, generated from Algorithm 1, for different values of the trade-off parameter p

This map is hard to read because too much information is displayed (*e.g.* in Western Europe). The map on the right presents the best organization generated by the algorithm for $p = 0.4$. Some groups are aggregated (*e.g* `Latin America` and `S. Africa`). They correspond to the groups in Fig. 2(a) that have a very low KL divergence. Other groups, that have a high information loss wrt their complexity reduction, are kept detailed. As p increases, higher-level groups are displayed, thus reducing the map complexity while saving the more information. This technique leads to multi-resolution maps that fit the variable distribution. For $p > 0.56$, only the total number of citations is displayed (full aggregation).

7 Conclusion and Perspectives

The design and debugging of complex MAS need abstraction tools to work at a higher level of description. However, such tools have to be built and exploited with the greatest precaution in order to preserve useful information regarding the system behavior and to guarantee that generated descriptions are not misleading. To that extent, this paper focuses on aggregation techniques for large-scale MAS and gives clues to estimate their quality in term of information content. They are applied to the geographical aggregation of international relations through the point of view of on-line newspapers. We show that, by combining information theoretic measures, one can give interesting feedback regarding geographical abstractions and build multi-resolution maps of the world that adapt the visualization complexity to the effective information content.

Future work will apply these techniques to other dimensions of the analysis: *e.g.* for temporal aggregation, thematic aggregation, multi-dimensional aggregation [16]. Besides this work, we are currently exploiting these techniques for performance visualization of large-scale distributed systems [17]. This kind of application shows that our techniques can be scaled up to 1 million agents.

Acknowledgement. This work is partially funded by the ANR-CORPUS GEOMEDIA project. We would like to thank C. Grasland, M. Severo, and T. Giraud for their work on this project; and L. M. Schnorr for its close collaboration.

References

1. Elmqvist, N., Fekete, J.: Hierarchical Aggregation for Information Visualization: Overview, Techniques, and Design Guidelines. IEEE Transactions on Visualization and Computer Graphics 16(3), 439–454 (2010)
2. Kullback, S., Leibler, R.: On Information and Sufficiency. Annals of Mathematical Statistics 22(1), 79–86 (1951)
3. Shannon, C.: A mathematical theory of communication. Bell System Technical Journal 27, 379–423, 623–656 (1948)
4. Gil-Quijano, J., Louail, T., Hutzler, G.: From Biological to Urban Cells: Lessons from Three Multilevel Agent-Based Models. In: Desai, N., Liu, A., Winikoff, M. (eds.) PRIMA 2010. LNCS (LNAI), vol. 7057, pp. 620–635. Springer, Heidelberg (2012)
5. Railsback, S.F., Lytinen, S.L., Jackson, S.K.: Agent-based Simulation Platforms: Review and Development Recommendations. Simulation 82, 609–623 (2006)
6. Búrdalo, L., Terrasa, A., Julián, V., García-Fornes, A.: A Tracing System Architecture for Self-adaptive Multiagent Systems. In: Demazeau, Y., Dignum, F., Corchado, J.M., Pérez, J.B. (eds.) Advances in PAAMS. AISC, vol. 70, pp. 205–210. Springer, Heidelberg (2010)
7. Tonn, J., Kaiser, S.: ASGARD – A Graphical Monitoring Tool for Distributed Agent Infrastructures. In: Demazeau, Y., Dignum, F., Corchado, J.M., Pérez, J.B. (eds.) Advances in PAAMS. AISC, vol. 70, pp. 163–173. Springer, Heidelberg (2010)
8. Sharpanskykh, A., Treur, J.: Group Abstraction for Large-Scale Agent-Based Social Diffusion Models with Unaffected Agents. In: Kinny, D., Hsu, J.Y.-j., Governatori, G., Ghose, A.K. (eds.) PRIMA 2011. LNCS (LNAI), vol. 7047, pp. 129–142. Springer, Heidelberg (2011)
9. Peng, W., Grushin, A., Manikonda, V., Krueger, W., Carlos, P., Santos, M.: Graph-Based Methods for the Analysis of Large-Scale Multiagent Systems. In: AAMAS 2009, pp. 545–552 (2009)
10. Iravani, P.: Multi-level network analysis of multi-agent systems. In: Iocchi, L., Matsubara, H., Weitzenfeld, A., Zhou, C. (eds.) RoboCup 2008. LNCS (LNAI), vol. 5399, pp. 495–506. Springer, Heidelberg (2009)
11. Grasland, C., Didelon, C.: Europe in the World – Final Report. ESPON Project 3.4.1, vol. 1 (2007)
12. United Nations Environment Programme: Global Environmental Outlook: environment for development, Nairobi, vol. 4 (2007)
13. Lamarche-Perrin, R., Vincent, J.M., Demazeau, Y.: Informational Measures of Aggregation for Complex Systems Analysis. Technical Report RR-LIG-026, Laboratoire d'Informatique de Grenoble, France (2012)
14. Csiszár, I.: Axiomatic Characterizations of Information Measures. Entropy 10(3), 261–273 (2008)

15. Deguet, J., Demazeau, Y., Magnin, L.: Element about the Emergence Issue: A Survey of Emergence Definitions. ComPlexUs 3, 24–31 (2006)
16. Lamarche-Perrin, R., Demazeau, Y., Vincent, J.M.: The Best-partitions Problem: How to Build Meaningful Aggregations? Technical report, Laboratoire d'Informatique de Grenoble, France (forthcoming, 2013)
17. Lamarche-Perrin, R., Schnorr, L.M., Vincent, J.M., Demazeau, Y.: Evaluating Trace Aggregation Through Entropy Measures for Optimal Performance Visualization of Large Distributed Systems. Technical report, Laboratoire d'Informatique de Grenoble, France, RR-LIG-037 (2012)

Multi-layered Satisficing Decision Making in Oil and Gas Production Platforms

Lars Lindegaard Mikkelsen[1], Yves Demazeau[2], and Bo Nørregaard Jørgensen[3]

[1] DONG Energy E&P
lalmi@dongenergy.dk
[2] CNRS, LIG Laboratory
Yves.Demazeau@imag.fr
[3] University of Southern Denmark
bnj@mmmi.sdu.dk

Abstract. From a control perspective, offshore oil and gas production is very challenging due to the many and potentially conflicting production objectives that arise from the intrinsic complexity of the oil and gas domain. In this paper, we show how a multi-layered multi-agent system can be used to implement a satisficing decision-making process for allocation of production resources. Through simulations using real-world production data, we illustrate that this satisficing decision-making process performs better than existing control systems applied on marginal fields, even though satisficing decision making often only provides near-optimal solutions.

Keywords: Multi-agent systems, Emergence, Satisficing, Multi-objective, Production Systems.

1 Introduction

The background for our research is oil and gas production at marginal (i.e. small in total oil and gas volume) fields in the Danish sector of the North Sea, more precisely the Siri Area. The Siri Area consists of three fields (Siri/Stine, Nini, and Cecilie). Oil and gas fields are typically owned by several companies (partners) to reduce the economic risk. Within the group of partners, one partner is normally the field operator. At the Siri Area, DONG Energy E&P is the operator. The operator has the daily responsibility for production and maintenance of the production platform's installations.

From a production perspective marginal fields are very challenging since they mature more rapidly, i.e. in the range of months, than larger fields that mature in the range of years. Furthermore, marginal fields may typically go through the full lifecycle from installation to abandonment in less than a decade. As a consequence of the rapid maturing, the production scheme of marginal fields has to be revised more frequently than those of regular fields. Simply applying the same relatively fixed production scheme as is used at regular fields would result in suboptimal production. The application of a relatively fixed production scheme is further challenged by the fact that the growing global request for oil and gas advances technological achievements

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 170–181, 2013.

which allow marginal fields to evolve beyond their original abandon point. This is also the case for the Siri Area, where the Siri production platform in 2004 became host for the first tieback project, such that it now consists of the main production platform Siri, three unmanned satellite platforms Nini, Nini East, Cecilie, and one subsea installation Stine. The focus of this paper is on the intrinsic complexity of the oil production platform and the oil production processes, with their indirect cross-production resource dependencies among production equipment and processes like water injection, gas handling, and tanker export. Indirect cross-production resource dependencies may result in unforeseen interactions, like bottlenecks and fluctuations, which cause the production scenarios to become very dynamic and complex. To avoid bottlenecks and fluctuations, process operators typically decrease the production throughput to increase process stability. By inspecting historical production logs at Siri, it is found that production throughput is frequently lowered and often goes on unnoticed for a long period before intervention happens, which results in unnecessary loss of production. This loss of production can be avoided if the production scheme is continuously adjusted to the dynamics of the field, a process which involves the three decision layers of a production platform: 1) The strategic production-decision layer handling the planning with goals in the range of weeks/days; 2) The tactical production-decision layer handling the allocation with goals in the range of days/hours; 3) The operational production-decision layer handling the local optimization with goals in the range of hours/minutes.

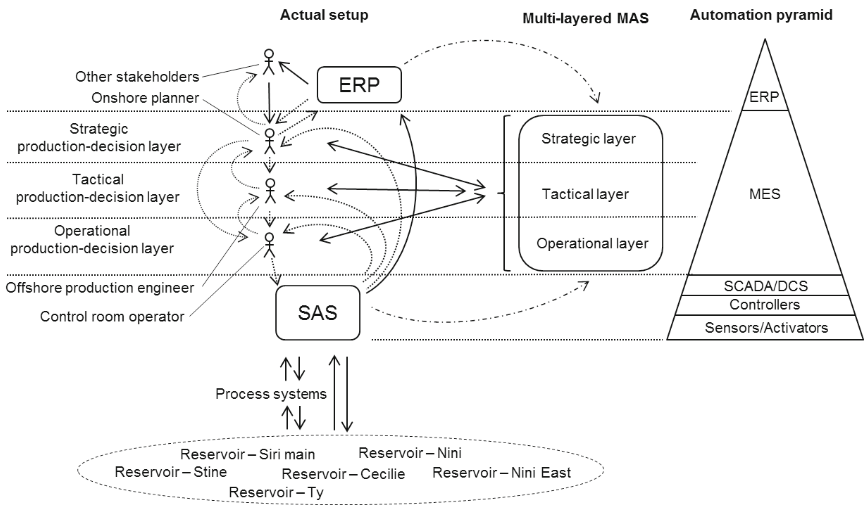


Fig. 1. Simplified data infrastructure at the Siri Area

Fig. 1 depicts a simplified data and control infrastructure diagram for the Siri Area, with the following abbreviations: ERP (Enterprise Resource Planning systems) and SAS (Safety and Automation System) [1]. Today, the data flow of integrated operation control systems is directed from the SAS system towards the ERP system and not vice versa. Data from the ERP system are primary used to generate new strategic, tactical, and operational plans, which from a production viewpoint makes the system relatively static. To support a flexible production scheme that automatically adjusts itself to the dynamics of the field, data must flow in both directions.

The rest of the paper is organized as follows: Section 2 presents the related state of the art. Section 3 provides the conceptual overview of our approach and describes its implementation. An experimental evaluation using real production data from the Siri Area is presented in section 4. Finally, section 5 concludes the paper.

2 State of the Art

Today, control systems for offshore oil and gas production are programmed in static control structures in well-proven industrial automation systems, i.e., DCS (Distributed Control System) or SCADA (Supervisory Control and Data Acquisition). Optimization is normally done offline, resulting in relatively fixed production schemes. Several internal DONG Energy E&P optimization studies conducted at the Siri Area have indicated that production throughput could be increased if the information systems at the decision layers, i.e., SAS and ERP systems, were better integrated thereby allowing a faster respond to changing field dynamics. The studies have also shown that a production throughput tends to decrease over time without any interconnected change in the related production constraints or objectives.

In general, studies addressing optimization in a mathematically strict sense is only seen in a very few studies, which address optimization of lift gas and slug mitigation, using Advanced Process Control (APC) and Model Predictive Control (MPC) as discussed by Bonavita et al. [2]; Artificial Intelligence in Petroleum Engineering discussed by Mohaghegh [3], and Real-Time Optimization (RTO) discussed by Bieker et al. [4]. The focus of these studies is optimization of subsystems and none of them, to the best of our knowledge, can handle online optimization of a complete installation. A few preliminary decision-support-control and distributed optimization systems by Ølmheim et al. [5] and Wartmann et al. [6] have also been tested in the oil and gas production domain, but only in simulation scenarios and only addressing confined parts of the process installation. None of the mentioned control approaches seems to have the ability to provide a high degree of flexibility for handling the changing high-level operational conditions at rapidly maturing marginal fields with limited data models, process data, and shared resources.

To meet the challenges of controlling a dynamic offshore oil and gas production environment in the range of minutes/hours/days (daily production issues) through months/years (field maturing issues), a multi-agent system with the characteristics of natural decomposition of action, perception, and distributed problem solving seems a promising approach as it provides the required flexibility. One of the first approaches in applying a multi-agent system to control production can be found in the ARCHON project [7] that proposed to encapsulate entities with cognitive layers [8]. Multi-layered multi-agent systems as we use them are not as such a new idea; some early work in this direction is the ASIC system developed by Boissier et al [9]. The ASIC architecture consists of three layers: command, adaptation, and decision. A three-layer model was also proposed in the work of Chappin et al., for conceptualization and formalization of agent behaviour in a socio-technical system for operational decision making in electricity markets [10]. Finally, Barbuceanu et al. followed a similar approach for organizing the supply chain in manufacturing as a network of cooperating, intelligent agents, each performing one or more supply chain operations [11]. In the context of control, the work of Sørensen et al. [12] shows how multi-agent systems can be used to prevent interactions among independent control objectives sharing resources in the same controlled environment, by searching for alternative resource allocation solutions that are acceptable to the requirements of all control objectives.

Building on state of the art, we propose a new application of multi-layered multi-agent systems to bridge the gap between SAS and ERP in complex offshore oil and gas production by developing a multi-layered multi-agent system that integrates the individual decision layers present in the automation pyramid of an oil and gas production platform. The open nature of multi-agent systems provides the necessary flexibility to meet the evolution of marginal fields, as new agents addressing changed production conditions can be added whenever the need arises. That is, the proposed multi-layered multi-agent system can adapt to new operational conditions, as it supports dynamical introduction and removal of control agents, each representing different production objectives, without the need to inspect or modify existing control agents. This dynamic control is possible as the infrastructure of the multi-layered multi-agent system takes responsibility for coordinating potential interactions among control agents dynamically.

3 The Approach and Its Key Criteria

In the production system domain, the focus is usually on "optimization", but for many real-world problems only limited models are available due to system complexity; so, in a mathematically strict sense, no optimization is performed, as there is no knowledge with regards to location of a global production optimum. Based on literature studies and in the light of the interviews given by oil-and-gas-production engineers, we argue that it is not "optimization" that is done today in oil and gas production, but merely manually tuning of process parameters based on human experience. Based on this observation, our approach aims to find solutions to allocation of production resources that satisfy the production objectives and constraints at a given time in the best possible way within a given time frame. Our approach is inspired by the economist Simon who introduced the concept of 'satisficing' as an approach to decision making [13]. The word 'satisficing' is a combination of the two words: 'satisfy' and 'suffice'. In [14], satisficing in the context of decision making is defined as:

> *"Examining alternatives until a practical (most obvious, attainable, and reasonable) solution with adequate level of acceptability is found, and stopping the search there instead of looking for the best-possible (optimum) solution".*

Satisficing decision making seems to be a useful approach in the development of complex control systems when no global model can be established to determine a global optimum. By using satisficing decision making, the aim is to find the best-possible solutions for allocation of resources to production processes that both satisfy and suffice all production objectives and constraints. Since decision making in the oil-and-gas-production domain is multi-layered, it is necessary to find satisficing solutions not only within each decision layer but also across all decision layers. As shown in Fig. 1, the decision-making process is scattered across the strategic, tactical, and operational layers. Ideally, decision-making within one layer should be coordinated with decision making within adjacent layers in order to find solutions that bring the whole production platform to a state of satisficing equilibrium.

Today, the flow of control typically propagates from the strategic layer through the tactical layer down to the operational layer, with no or very little feedback to upper layers, in case the lower layers cannot meet the demands of the decisions made at the upper layers. That is, any assumptions an upper layer may hold about the effects of its decisions at lower layers may be broken without the upper layer knowing about it, which may lead to suboptimal or even wrong subsequent decisions at the upper layer. Those unnoticed broken assumptions between multi-layered decision layers can be avoided, if they are made explicit by providing feedback from lower layers to higher layers. Such a feedback mechanism can be established based on Jackson's work on problem frames [15]. We use Jackson's concept of entailment relations [16] to discover broken assumptions. An entailment relation is a tuple ($S, W \vdash R$), where:

1. Specifications S are implemented as a computer program.
2. The World W or the set of domain properties as seen by S.
3. $\vdash$ is the semantic for entailment.
4. R is the user Requirements.

As the entailment relation ($S, W \vdash R$) is defined for a single problem context, we have to extend the concept of entailment to cover nested sub-problems in order to use it for a multi-layered decision-making process. This can be done by defining a nested entailment relation that propagates information from lower decision layers to upper decision layers. In the nested entailment relation the world W at the upper layer $L1$ will be given by an entailment relation ($S, W \vdash R$) at the lower layer $L2$. I.e., in the entailment relation ($S_{L1}, W_{L1} \vdash R_{L1}$) at layer $L1$, W_{L1} will be given by an entailment relation ($S_{L2}, W_{L2} \vdash R_{L2}$) at layer $L2$, and W_{L2} at layer $L2$ will be given by an entailment relation ($S_{L3}, W_{L3} \vdash R_{L3}$) at the lower layer $L3$ and so on, until entailment is terminated when the bottom layer is reached. When implementing the concept of entailment in a multi-agent system, each specification in the set S becomes an agent, the world W becomes the agents' world model and the entailment $\vdash$ of requirements R becomes the goals of the agents. In a control context, W usually expresses inputs and outputs of the control system. When the chain of nested entailment relations holds, we define the system to be in a state of satisficing equilibrium. The possible size of the satisficing equilibrium state space depends on the flexibility in the requirements R.

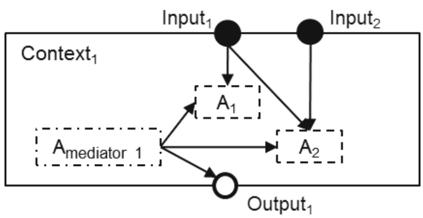


Fig. 2. Negotiation context

Agents belonging to the same entailment relation are grouped into a negotiation context with one mediator agent responsible for the negotiation process. Fig. 2 depicts a single negotiation context (solid rectangle) with agents A_1 and A_2 (dashed rectangles). The world *W* is represented by Input_1, Input_2 and Output_1. The arrows indicate the direction of data flow. There is at least one negotiation context for each decision layer.

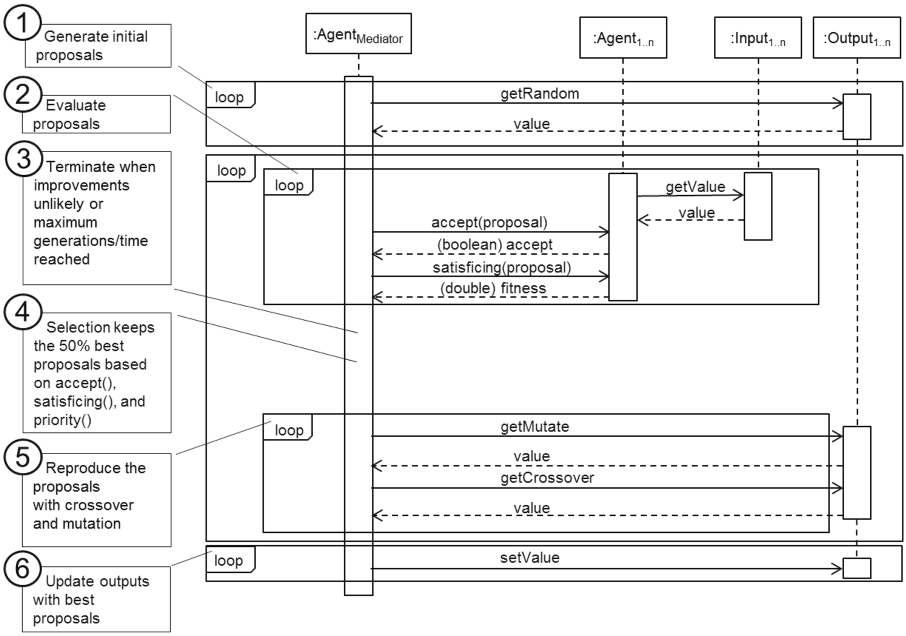


Fig. 3. UML sequence diagram of negotiation process

The negotiation process is an incremental process that converges towards a satisficing solution, if one exists. In searching for a satisficing solution, the mediator uses a genetic algorithm. Implementation details concerning the genetic algorithm can be found in [12]. The negotiation process is divided into six steps as shown in Fig. 3.

1. The mediator agent generates an initial set of proposals (e.g. 200) for allocation of production resources.
2. The mediator agent presents each proposal to the agents by message *accept* and they reply whether they can accept the proposal. They furthermore respond to a message *satisficing* about how satisfied they are with each specific proposal. Satisficing is here expressed as a percentage to which the proposal fulfils the agent's goals.
3. The mediator agent continues the negotiation process as long as it has not converged towards a solution and the end of the control loop's time period is not reached. The negotiation process is said to have converged, and a satisficing solution found, when no new proposal can be generated, that is any better than a previous proposal. When the negotiation process terminates, the mediator agent jumps to step 6.
4. The mediator agent selects the best 50% of the proposals based on acceptance, priority and the returned fitness values.
5. The mediator then generates the missing 50% proposals using the genetic algorithm's crossover and mutation functions. The mediator then loops to step 2.
6. Finally the mediator updates all outputs with accepted solutions.

In case no satisficing solution could be found, the entailment relation of the negotiation context is said to be broken. Broken entailments are typically caused by allocation conflicts over shared production resources. E.g. in Fig. 2, $Output_1$ is shared by agents A_1 and A_2. Many of these resource-allocation conflicts may emerge, because agents by default are considered equally important. However, in any non-trivial control system the importance of individual agents may change depending on the actual operational state. This state-dependent change in agents' importance is handled by supporting dynamic prioritization of individual agents. Important agents are given higher priority than less important agents. By default the priority of all agents is set to 5. In the current implementation, we have chosen to use a priority range from 1-10 (1 = highest and 10 = lowest). In selection of the best 50% of the proposals, the mediator favours proposals accepted by agents with higher priorities.

Furthermore, to acknowledge the fact that some agents' requirements may be more critical, from a safety point of view, than others, agents' requirements can either be modelled as 'hard' or 'soft'. A 'hard' requirement is a requirement that always have to be satisfied, and a 'soft' requirement is a requirement that it is desirable to satisfy. For instance, in the oil and gas production domain the complete production platform is protected by the safety system, so the outer bounds of the solution space are set by the safety system. The constraints in the safety system have to be mapped to hard requirements to avoid production shutdowns, as eventual shutdowns are very expensive due to slow start-ups and thereby lost production. Examples of safety system constraints can be minimum/maximum temperatures, tank-levels, pressures etc. Another group of hard requirements is related to business logic/policies like in the Siri Area where each of the three oil fields (Siri/Stine, Cecilie, and Nini) are processed in separate separators even though a mixed field mode could give a better throughput. Production-related issues like optimizing flaring over water injection are 'soft' requirements. The set of 'hard' requirements confines the search space for finding a satisficing solution. Hence a satisficing solution to the allocation of production resources is a solution that fulfils all 'hard' requirements and provides the best-possible solution for all 'soft' requirements.

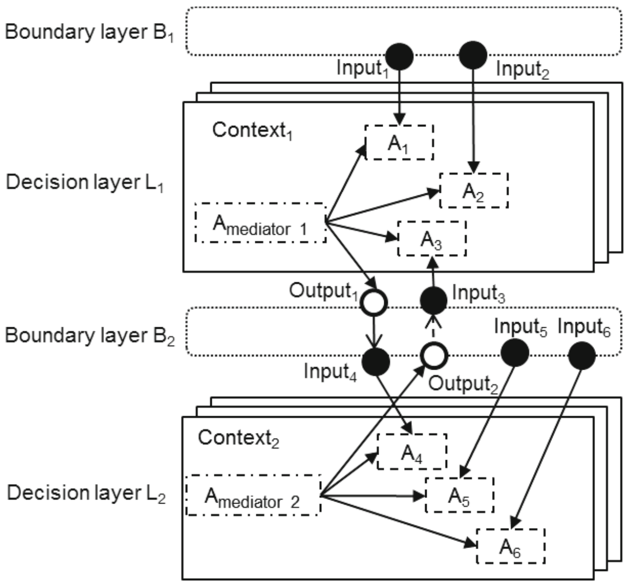


Fig. 4. Agent interactions between negotiation contexts on different layers

Each decision layer may contain one or more negotiation contexts that are connected to negotiation contexts in adjacent decision layers through the inputs and outputs of a boundary layer, as shown in Fig. 4. The boundary layer provides the necessary feedback mechanisms for delegating broken entailment relations between decision layers, thereby ensuring that the system is in a constant search for a satisficing equilibrium. Using the inputs and outputs of the boundary layer $B_{2,}$ the feedback mechanism links the entailment relation (S_{L2}, $W_{L2} \vdash R_{L2}$) for Context$_2$ at decision layer L_2 to the world W_{L1} of Context$_1$ at decision layer L_1. In Fig. 4 the negotiation Context$_1$ at decision layer L_1 is connected to the negotiation Context$_2$ at decision layer L_2. When the negotiation process of Context$_1$ terminates, mediator $A_{Mediator\ 1}$ passes on the best solution to decision Layer L_2 in the form of Output$_1$. Output$_1$ is through boundary layer B_2 provided as input Input$_4$ at decision layer L_2. Mediator $A_{Mediator\ 2}$ now seeks for a solution at decision layer L_2, and feedback to decision layer L_1 is through a domain property of the world W_{L1} (marked with a dashed arrow between Output$_2$ and Input$_3$) in the boundary layer B_2. Hence, agent A_3's world W entails Context$_2$ based on Input$_3$. Thus, in case the entailment relation (S_{L2}, $W_{L2} \vdash R_{L2}$) at decision layer L_2 is broken, due to the value of Input$_4$, it is propagated back to decision layer L_1 through the nested entailment relation between Output$_2$ and Input$_3$.

The design of the feedback mechanism also ensures propagation of any change in the operational state of $Context_2$, for instance in case agents are prioritized, inputs change, a new solution is found, etc. As an example, if agent A_4 is given a higher priority than A_5, this may impact $Output_2$; the feedback via the domain property of W_{L1} will impact A_3. A_3 will seek to influence $Output_1$; this process will continue until a satisficing equilibrium is found or a conflict is identified. In the next section, the experiment illustrates the use of both feedback and priority. To ensure stability of the multi-layered negotiation process in search for a satisficing equilibrium, we use the basic rules of thumb on closed-loop control, expressing that the inner loop (lower decision layer) is twice as fast as the outer loop (upper decision layer).

4 Experiments

To validate our approach, we have chosen the export-to-tanker scenario at the Siri platform, as this scenario involves all three decision layers. The experiment extends our previous findings [17, 18]. In the export-to-tanker scenario, the oil is exported to a shuttle tanker from a temporary storage tank on the seabed. Due to limited electrical power resources at the Siri platform, one of the major power consumers has to be stopped during the export-to-tanker scenario, i.e., either a gas compressor or a water injection pump. The gas compressors are used to handle produced gas either for use as fuel or lift gas (lift gas is used to get the wells to flow due to low pressure in the reservoirs). Water injection is used as pressure support in the thin sandstone production layers in the reservoirs in order to maintain an economically-feasible production. The water injection system consists of three 2 MW pumps. Fig. 5 shows negotiation contexts (solid rectangles) that are directly involved in the experiment. Names of negotiation contexts at the strategic layer are related to resource allocation concerns, whereas at the tactical layer they are related to systems, and at the operational layer they are related to production equipment.

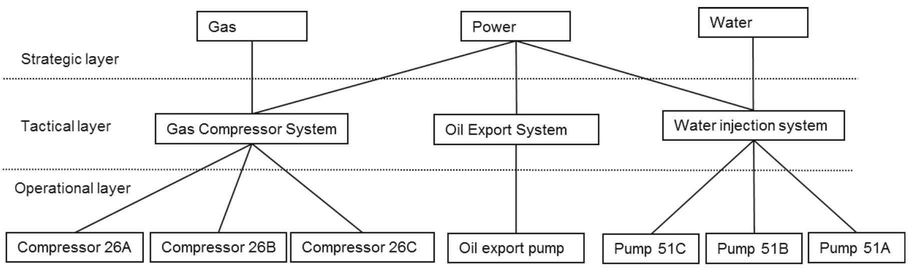


Fig. 5. Experiment negotiation context diagram

The export-to-tanker scenario is triggered by the *Oil export system* at the tactical layer, and by requesting power from the *Power* negotiation context. The *Power* negotiation context contains the following agents: *Ensure power for water injection* (priority 6), *Ensure total power* (priority 3), *Ensure power for gas compression* (priority 4), *Maximize total power allocated* (priority 6), *Ensure power for export* (priority 5). The *Power* mediator agent starts a negotiating process to find a new power plan. As the flare is to be kept at a minimum due to the environmental impact and the appertaining regulations, the agent *Ensure power for gas compression* has been assigned with a higher priority than the agent *Ensure power for water injection*. Based on the agents' priorities, we expected that a water injection pump would be stopped during the export-to-tanker scenario.

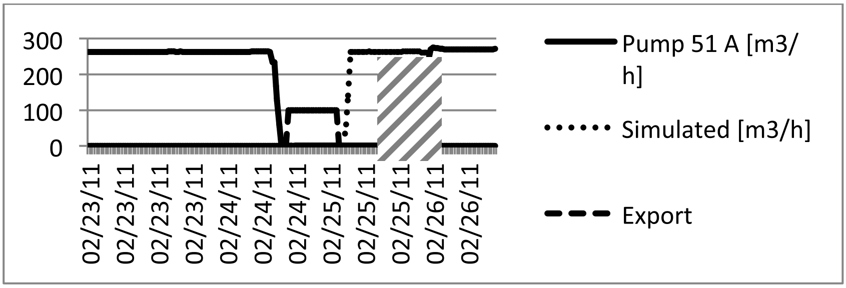


Fig. 6. Real production data - Export to tanker

Fig. 6 shows real production data for the export-to-tanker scenario executed manually at the Siri platform on 24-01-2011. The dashed line indicates when the export-to-tanker scenario took place. During this export the water injection pump 51A (solid line) was manually stopped. The manual start of water injection pump 51A was postponed by a day (solid line) without a technical reason, resulting in an unnecessary loss of production. A simulation using our approach shows that the water injection pump would have been automatically started one day earlier (marked with the dotted line). This earlier start gives a better water injection performance than the manually operated system. The increase in the water injection volume is marked by the hatched area. Our simulation run of the real production data is shown in Fig. 7. The three layers from Fig. 5 and their respective negotiation contexts are mirrored in the GUI.

In Fig. 7 agents are coloured depending on how satisfied they are with a solution. Satisfied agents are green (dark grey), and dissatisfied agents have shades ranging from yellow (light grey) to dark orange (medium grey). In the simulation, the agent *Ensure power for water injection* turns orange (medium grey) when the export-to-tanker scenario starts, as the agent *Ensure power for export* has higher priority. Similarly, the agent *Ensure pump capacity for Siri* turns orange as the power allocation for the Water injection system is insufficient to maintain full pump capacity. Later on, when the export-to-tanker scenario stops, the *Power* negotiation context will again allocate the necessary power for the water injection system, and the agents *Ensure power for water injection* and *Ensure pump capacity for Siri* will turn green again.

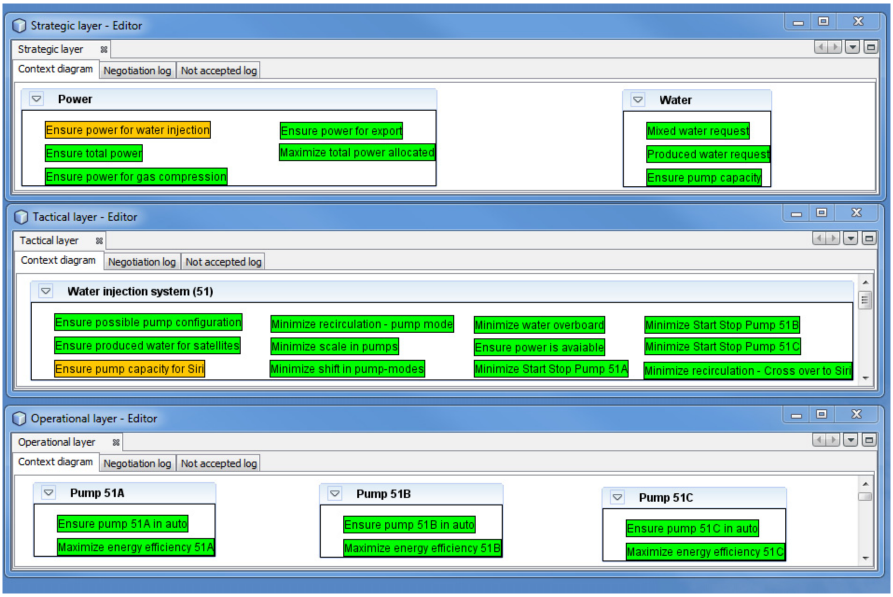


Fig. 7. Simulation's run for export-to-tanker scenario

5 Conclusion

In this paper, we have presented a new application of multi-layered multi-agent systems for supporting decision making across the three layers of control in offshore oil and gas production. The proposed approach provides a new level of flexibility that meets the need for dynamic evolution of marginal oil and gas fields.

We have shown that a satisficing decision-making process implemented as a multi-layered multi-agent system can perform better than manually controlled systems, as is currently the state of the art within the oil-and-gas-production domain. Hence, we believe that the proposed approach possesses the capability to face the continuously changing operational conditions of marginal oil and gas fields in the North Sea.

From an architectural perspective the proposed multi-layered multi-agent-system approach is not bound to the oil and gas domain, and it seems reasonable that the approach can be mapped to other control domains which possess a similar layered structure for decision making.

Acknowledgement. Noreco and RWE, our partners in the Siri Area, are greatly acknowledged for the permission to publish this paper. The partners do not take responsibility of neither the contents nor the conclusions. The authors would furthermore like to thank the anonymous reviewers for their valuable suggestions to improve the paper.

References

1. NORSOK Standard I-002: Safety and automation system (SAS). Rev. 2 (2001)
2. Bonavita, N.: Performance Excellence in the Upstream Industry. In: 10th Mediterranean Petroleum Conference, Tripoli, Libya (2008)
3. Mohaghegh, S.D.: Recent Development in Application of Artificial Intelligence in Petroleum Engineering. Journal of Petroleum Science and Engineering, 239–260 (2005)
4. Bieker, H.P., Slupphaug, O., Johansen, T.A.: Real-Time Production Optimization of Offshore Oil and Gas Production Systems - A Technology Survey. In: SPE Intelligent Energy Conference and Exhibition, Amsterdam, NL (2006)
5. Ølmheim, J., Landre, E., Spillum, Ø., Hepsø, V.: Decision Support and Monitoring Using Autonomous Systems. In: SPE Intelligent Energy Conference and Exhibition, Utrecht, NL (2010)
6. Wartmann, M.R., Gunnerud, V., Foss, B.A., Ydstie, B.E.: Distributed optimization and control of offshore oil production: The intelligent platform. In: FOCAPO 2008, Boston, USA (2008)
7. Jennings, N.R., Mamdani, E.H., Corera, J.M., Laresgoiti, I., Perriolat, F., Skarek, P., Varga, L.Z.: Using ARCHON to develop real-word DAI applications. IEEE Expert 11(6), 64–70 (1996)
8. Demazeau, Y.: From Cognitive Interactions to Collective Behaviour in Agent-Based Systems. In: 1st European Conference on Cognitive Science, Saint-Malo, France, pp. 117–132 (1995)
9. Boissier, O., Demazeau, Y.: ASIC: An Architecture for Social and Individual Control and its Application to Computer Vision. In: Perram, J., Müller, J.P. (eds.) MAAMAW 1994. LNCS (LNAI), vol. 1069, pp. 135–149. Springer, Heidelberg (1996)
10. Chappin, E.J.L., Dijkema, G.P.J., van Dam, K.H., Lukszo, Z.: Modelling strategic and operational decision-making—an agent-based model of electricity producers. In: The 2007 European Simulation and Modelling Conference, pp. 356–363 (2007)
11. Barbuceanu, M., Fox, M.S.: The architecture of an agent based infrastructure for agile manufacturing. In: Proceedings of IJCAI (1995)
12. Sørensen, J.C., Jørgensen, B.N., Klein, M., Demazeau, Y.: An agent-based extensible climate control system for sustainable greenhouse production. In: Kinny, D., Hsu, J.Y.-j., Governatori, G., Ghose, A.K. (eds.) PRIMA 2011. LNCS (LNAI), vol. 7047, pp. 218–233. Springer, Heidelberg (2011)
13. Simon, H.A.: Rational choice and the structure of the environment. Psychological Review 63(2), 129–138 (1956)
14. Definition source, `http://www.businessdictionary.com/definition/satisficing.html`
15. Jackson, M.: Problem Frames: Analysing & Structuring Software Development Problems. Addison-Wesley (2001) ISBN-10 0-201-59627-X
16. Jackson, M.: The World and the Machine. In: Proceedings of the 17th International Conference on Software Engineering, pp. 283–292 (1995)
17. Mikkelsen, L.L., Jørgensen, B.N.: Towards Next Generation of Smart Fields Using Intelligent Online Multi-Objective Control. In: POSE 2012, Doha, Qatar, SPE 156946 (2012)
18. Mikkelsen, L.L., Jørgensen, B.N.: Applying Multi-agent Systems to Multi-objective Simulation and Control for Offshore Oil and Gas Production. In: RACS 2012, San Antonio, Texas, USA, pp. 376–382 (2012)

A Security Response Approach Based on the Deployment of Mobile Agents

Roberto Magán-Carrión, José Camacho-Páez, and Pedro García-Teodoro

Department of Signal Theory, Telematics and Communications,
Faculty of Computer Science - CITIC, University of Granada
{rmagan,josecamacho,pgteodor}@ugr.es

Abstract. This paper introduces a response mechanism to improve the tolerance against security threats in MANET environments. The mechanism is started after detecting the existence of nodes with malicious behavior, and is based on the use of one or more mobile agents to improve the connectivity of the network. This way, in the event of the detection of a malicious node (e.g. a *selfish* node or a *dropper* node), an agent is employed to maximize the overall connectivity of the network. Every agent acts as a relaying node within the MANET and it is automatically positioned according to a particle swarm optimization (PSO) process. This paper represents a work in progress. However, the promising results obtained show the good suitability of the approach to improve the survivability of the network from a security perspective.

Keywords: agent, PSO, detection, malicious, MANET, response, survivability, tolerance.

1 Introduction

In the context of ad hoc networks, mobile ad hoc networks (MANET) have a several special characteristics: lack of a fixed infrastructure, dynamic changing topology, resource constraints and limited physical security, among others [1]. In this kind of networks, the communication between nodes is restricted and depends of their coverage range. Therefore, not all the nodes are directly connected, and thus a multi-hop relay-based scheme is needed for end-to-end transmissions. Some possible applications of MANETs include military scenarios, e.g. soldier communications in the battlefield; emergency rescue, e.g. earthquake or fire disasters when the fixed communication infrastructures are no longer available.

Compared with traditional wired networks, MANETs are much more vulnerable to attacks due to the limited energy of nodes, thus avoiding the use of complex security solutions, the wireless transmission medium, which makes eavesdropping easier, the lack of management and control unit and the implicit mobility of these environments. Attacks like *blackhole*, *sinkhole*, *dropping* or malicious behaviors as *selfish*, are specific for MANETs [2]. These inherent threats have an obvious and high impact over the network performance, since nodes need to send information through intermediate neighbors that could be attackers. Thus, security mechanisms to strengthen the services provided are needed.

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 182–191, 2013.

Deploying efficient security systems to reduce risks and threats by providing proper mechanisms to maximize the network performance is also required. This will raise the network survivability, which is defined as *"the ability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures or accidents"* [3].

In this context, the present work proposes a multiagent-based system aimed at enhancing the connectivity between nodes in the network in the presence of nodes with malicious behaviors. Hence, a tolerant and resilient network is obtained. The response mechanism presented here is based on the work of *Dengiz et al.* [4], where the authors improve the network performance by maximizing the connectivity and the amount of flows transmitted. They make use of the particle swarm optimization (PSO) algorithm and the future user node locations (kinematic predictions), to situate the agent nodes in optimal positions to maximize the overall connectivity.

The principal contribution of this paper is the use of mobile agents as a response mechanism for improving security in MANETs. This way, in the event of the detection of malicious nodes in the environment, the corresponding worsening of the network performance is mitigated by deploying some agents in charge of relaying packets and thus improving overall coverage and connectivity. Even though this paper represents a work in progress, and thus much more experimentation is in due, the preliminary experimental results obtained show the promising performance of our approach.

The rest of the paper is organized as follows. Section 2 presents some relevant works about multiagent systems in general, and security related ones in particular. Section 3 introduces a discussion about the reference work and the novel system proposed here. Some experimental results and simulation to corroborate and validate the system efficacy are described in Section 4. Finally, Section 5 summarizes the principal conclusions and remarks of this work as well as future research directions.

2 Related Work

There are several proposals in the literature on the use of mobile agents in ad hoc wireless networks addressing different kinds of challenges: network connectivity and node optimization positioning, improvement of network QoS parameters, energy optimization and security issues. In [5] the locations of agent nodes are optimized by means of a PSO algorithm to maximize the connectivity between user nodes and a control node. A PSO algorithm is also used in [4] enhanced with a kinematic prediction of user nodes motion following the scheme of model predictive control (MPC), with the aim of maximizing the connectivity and flow transmission in MANETs. Furthermore, in military scenarios, a supervisor mobile agent trajectory is optimized according the deployed positions of the user nodes, thus maximizing the connectivity between the control node and the arranged user nodes [6].

Ant colony optimization (ACO) and bee colony optimization (BCO) algorithms are used to improve QoS parameters and for energy optimization. Packet delivery ratio and end-to-end delay are the focus in [7]. In that reference, an ACO scheme is used to find out the shortest path both in routing discovery and maintenance phases. Similarly, the authors in [8] improve the network bandwidth by using an ACO algorithm. Here, a number of ants are in charge of discovering the best routes according to the destination distance, the available bandwidth and the queue of the nodes. An efficient routing algorithm is proposed in [9], where the routes are selected by employing bees to select the path with least energy consumption requirements.

Deploying attack detection (recognition) and event response (recovery) mechanisms constitute key issues for network survivability [3]. Non-legitimate event detection in networks is an aspect that has been recurrently studied in the specialized literature. On the other hand, response mechanisms try to solve the non-legitimate events detected in order to guarantee the continuity of the network and the affected services. Nevertheless, in the multiagent system field just a few works have been developed to address security detection and response issues, and these are limited to the use of software agents. In [10] an agent node is created and sent from the sender to the destination crossing the suspected node. If the agent never comes back, the suspected node is concluded to be a *grayhole* or a *blackhole* node. Unlike the previous work, in [11] the agent records the amount of packets received and forwarded by each node along the path. If the agent detects that the forwarded and received packets ratio is under a fixed threshold, the node is labeled as malicious. Then, a report is sent to the sender. A scheme imitating the human immune system is proposed in [12]. There exists an immune agent (IA) that is distributed along the network. The IA is in charge of detecting, classifying, isolating, and recovering if needed. A node that exceeds a certain number of attacks launched will be isolated. Reference [13] introduces a similar scheme, where the nodes are monitored and, if necessary, isolated by using two types of mobile agents: detection agent and counterattack agents. The first ones are in charge of detecting malicious behaviors, while the second ones will surround and isolate the invaders. In [14] the dynamic source routing protocol (DSR) for MANETs is modified by attaching two agents to each network node: a monitoring agent (MOA) and a routing agent (ROA). The first one monitors the node behavior to assign a trust value. The trust values are spread throughout the network into the route request packets. Afterwards, the ROA agent selects the trustworthiness route discarding others with less trust level. Thus, the nodes with low trust value are isolated.

In what follows we propose the use of physical and mobile agents as a response mechanism. Based on the pre-existence of a detection module for malicious behaviors, our approach consists of the deployment of one or more agents to solve the loss of "coverage" due to malicious nodes. The agents, acting as relaying nodes, will allow to improve the overall connectivity of the network.

3 MARS: Mobile Agent Response System

As mentioned above, the proposed system is inspired by the work of *Dengiz et al.* work [4], in which the connectivity of a MANET is improved by using mobile agent nodes. In this section, a brief explanation of that approach is first presented. A description of the specific response system proposed here, named MARS, is afterwards provided.

3.1 Connectivity Maximization Using PSO and Future Motion Predictions

Two types of nodes are involved in [4]: user nodes and agent nodes. User nodes are final nodes demanding some given network service, while agent nodes try to guarantee that user nodes are receiving the best network service as possible by maximizing the overall connectivity. The connectivity is related to the coverage range. Two nodes are accessible or connected (that is, there is a link between them) if the Euclidean distance between them is less or equal to R, where R is the coverage range of a radio node.

Basically, the authors in [4] suggest two objectives: to maximize the overall connectivity of the network, and to optimize the flow transmission. This optimization process is achieved by using the PSO algorithm [15] and several optimization functions. Two important and particular entries of the PSO algorithm are: the future motion predictions of user nodes for a specific prediction time horizon $(t + H)$, and the best solution previously obtained. Afterwards, a comparison among several possible problem solutions (*particles*) is made. The different particles in the same PSO run are specific network distributions where the user node locations at $t + H$ are the same and the agent nodes positions are modified by increasing or decreasing its velocity and direction values. When the PSO optimization process is finished, the algorithm returns the best locations of each agent node to maximize the overall connectivity and flow transmission of the network at a given time instant.

There are three optimization functions involved. With the first, O_1, the global network connectivity is evaluated. This can be obtained from:

$$O_{1t} = \frac{2 \times \sum_{i,j \in UN_t : j > i} Z_{ijt}}{UN \times (UN - 1)} \qquad (1)$$

where UN is the number of user nodes and $Z_{ijt} = 1$ if there exists an available (either single or multi-hop) path connecting the ith and jth user nodes at time t. Otherwise, $Z_{ijt} = 0$.

A second function, O_2, maximizes the flow transmission by improving the weakest link, in flow terms, of the network, which will enhance the overall network performance. O_2 is only evaluated when several possible solutions of the optimization (several particles under evaluation) represent completely connected networks, that is if $O_1 = 1$ in all of them. For disconnected networks, a third function O_3 is considered, which measures the distance from each agent to the imaginary middle point between non connected partitions of the network. In summary, the solution with a higher O_1 value is the best solution. If there are several completely connected networks in the solution of the optimization, the one with higher O_2 value is selected. However, if there are disconnected networks with the equal O_1 values, the one with lower O_3 value is chosen.

The optimization algorithm is iteratively repeated over the time, the agents being dynamically positioned at their best locations step by step. More details about the entire process can be found in the reference paper [4].

3.2 System Description

As commented in Section 2, there are few research proposals about response solutions in MANET networks, specially involving multiagent systems. Instead, most of them are merely related to the detection of malicious behaviors.

In this context, MARS is intended to establish a response mechanism for security alarms in MANETs by using mobile agents. MARS is conceived to be a tolerant mechanism against attacks such as *selfish*, where a node has an egoistic behavior not forwarding packets to preserve its own resources (e.g. battery life); or *dropping*, where a node drops the received packets instead of forwarding them.

Both types of attack, *selfish* and *dropping*, among others, have similar consequences on network performance. This way, these kinds of malicious nodes will prevent other nodes to communicate with each other. Using MARS, such attacks can be dealt with:

1. A monitoring system is deployed to detect this kind of behaviors. See [16] for an example of such a system. The design of the monitoring system is out of the scope of this paper.
2. Once one or more malicious nodes are detected, the MARS response mechanism is triggered. A set of mobile agents are dynamically launched, which present two main features. First, they act as mere relaying nodes to solve the decrease in connectivity in the network due to the appearance of malicious nodes. Second, the base optimization algorithm proposed in [4] is executed to determine the best positions of the agents to maximize the overall connectivity over the time.

To illustrate how MARS works, Figure 1 depicts different network situations. Initially, several user nodes (solid circles) are randomly distributed throughout a given network area, where there is also one malicious node (inverted triangle) and one agent node (solid square). At the beginning, in Figure 1(a), the malicious node works as a normal node. Afterwards, in Figure 1(b) the attack is in progress, so that the overall connectivity is broken and two separated networks are obtained. Figures 1(c) and 1(d) show the coverage recovery process. The agent, A1, is approaching to its optimal position (according to PSO), thus making possible the connection among the previously disconnected user nodes. In this case, the use of one single agent cannot provide full connectivity between user nodes due to their motion and the coverage range. Nevertheless, the position of A1 is optimally computed in order to connect the maximum number of nodes, as shown in Figures 1(e) and 1(f).

4 Experimental Results

This section is devoted to study the performance of our security response/tolerance system. For that, a set of experiments in a simulation scenario with Matlab are carried out. The main features of the scenario are:

1. The network area is 5m × 5m.
2. The coverage range R=1m, thus assuring disconnection among nodes.
3. The prediction horizon H=4, because it is an optimal value that offers better connectivity values in accordance with [4].

In order to evaluate the evolution of the connectivity of the network under various situations with respect to the number of final user nodes (UNs), number of malicious nodes (MNs), and number of agent nodes (ANs), ten different combinations are proposed:

1. With no attacks, that is under normal operation of the user nodes. In this case, we are going to obtain the connectivity for 10, 20, 30 and 40 UNs.
2. In the presence of attacks. In this case, the connectivity is obtained for 40 UN, while the number of MNs is taken to be equal to 1, 5 and 10.
3. With attacks and ANs to solve them. The study of the connectivity is done for 40 UNs, 10 MNs and the number of ANs equals to 1, 5 and 10.

Each connectivity analysis involves 25 repetitions for each experiment, where different initial random distributions of UNs and ANs in the network area are considered. Figures 2, 3 and 4 show the results obtained from our experimentation. As expected, O_1 increases in Figure 2 with the number of user nodes, since effective links will be established among them due to the fact that the global area size does not change. In other words, UNs are nearer to each other. The negative effect of the MNs in the network connectivity is illustrated in Figure 3, which shows how this connectivity decreases as the number of MNs increase. Finally, Figure 4 shows that the deployment of ANs in the environment, once the MNs are identified, contributes to the recovery of the connectivity of the overall network. On the one hand, the connectivity increases with the number of ANs. On the other hand, the connectivity with 10 ANs and 10 MNs is even higher than when there is no MNs in the MANET. This shows the effectiveness of the approach, and it is the result of the optimization of the ANs location.

Figure 4 shows that the deployment of ANs in the environment, once the MNs are identified, will contribute to recover the connectivity of the overall network. On the one hand, the connectivity increases with the number of ANs. On the other hand, it is important to remark the fact that the overall connectivity of a network for a given number of user nodes plus agent nodes is higher than when all the nodes are user nodes. The reason for that is simple, since the ANs are intelligent in the sense that they try to be positioned to maximize the global connectivity.

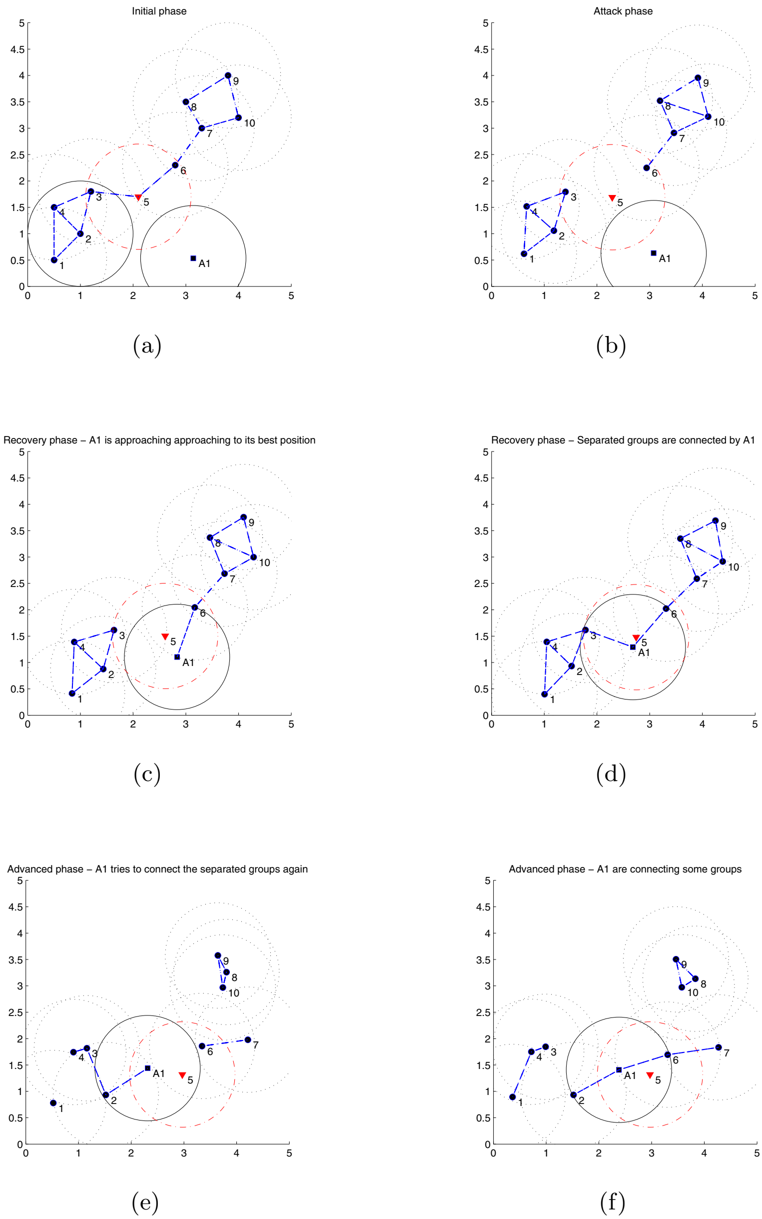


Fig. 1. Connectivity maximization and recovery process. Initial phase where a malicious node (inverted triangle) is performing the forwarding process together with the user nodes (solid circles) (a). Two separated groups of nodes result when the attack is carried out (b). Motion of the agent A1 when approaching to its optimal location recover connectivity and thus communications (c) (d). The agent is finally positioned to connect the maximum number of nodes (e) and (f).

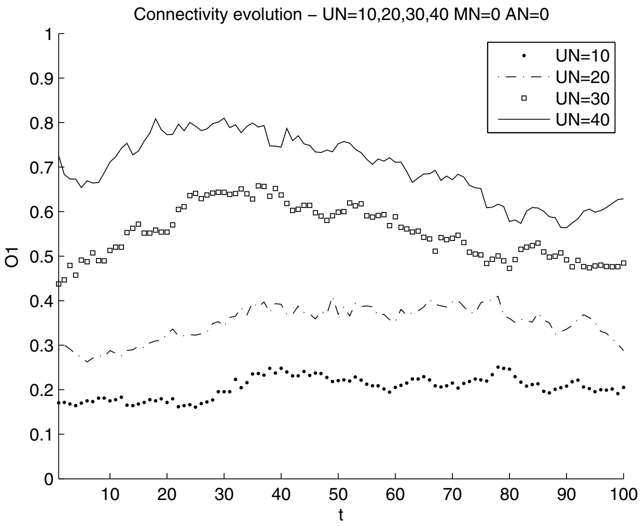


Fig. 2. Connectivity evolution along 100 time instants in normal conditions without influence of attacker or agent nodes. As expected, the connectivity increases as the number of user nodes, UN, becomes higher.

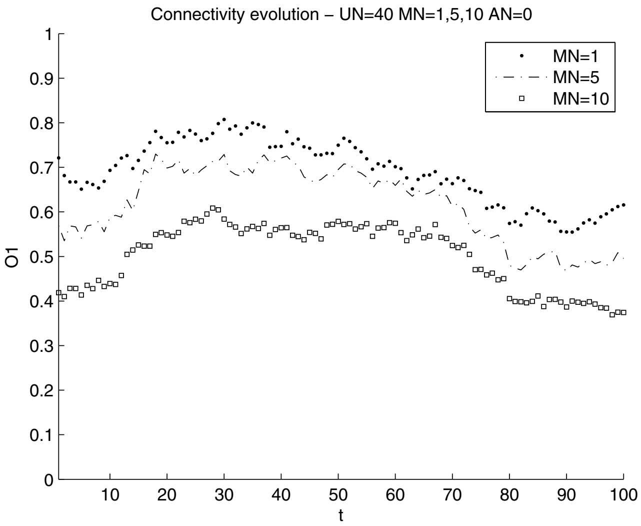


Fig. 3. Connectivity evolution along 100 time instants under influence of attacker or malicious nodes (MN). As expected, the connectivity becomes lower as the number of MNs increases.

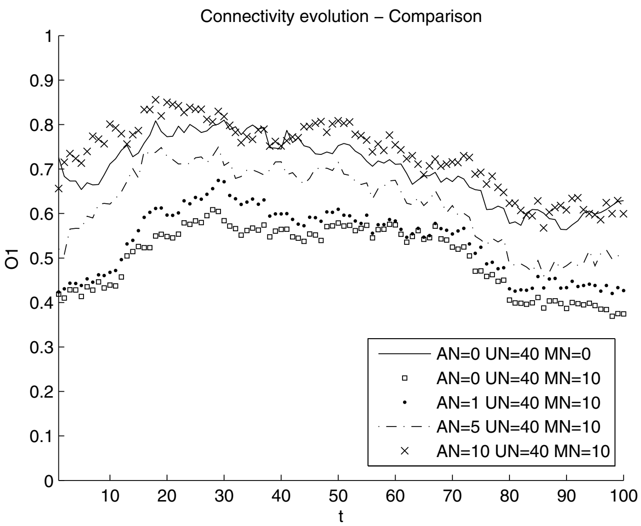


Fig. 4. Connectivity evolution along 100 time instants under influence of both malicious and agent nodes. The agents contribute to improve the connectivity, so it is raised as the number of ANs does. Also a comparison with two previous cases is done, in order to show the connectivity improvements gained by the agent nodes position optimization.

5 Conclusions and Future Work

In this paper a novel response/tolerance approach for security threats in MANETs is proposed. It is based on the use of mobile agent nodes, which are launched after detecting the existence of malicious nodes in the environment that decrease the connectivity of the network. A positioning optimization procedure is carried out to determine the best positions of agents over the time.

The experimental results obtained show the improvement in connectivity and thus the tolerance and survivability exhibited by the network operation when confronted to security attacks, when our approach is considered. Nevertheless, further work should be performed to strengthen the current proposal. Some issues may be mentioned in this line. First, alternative parameters to that of connectivity may lead the positioning of the nodes depending on different objectives established for the network. Second, the agent nodes can operate in different ways depending of other various types of attacks. Third, the response/tolerance scheme could be used as a feedback element to strengthen the security design of the network.

Acknowledgment. This work has been partially supported by Spanish MICINN through project TEC2011-22579 and by the FPU P6A grants program of the University of Granada.

References

1. Wu, B., Chen, J., Wu, J., Cardei, M.: A Survey of Attacks and Countermeasures in Mobile Ad Hoc Networks. Springer, US (2007)
2. Ehsan, H., Khan, F.A.: Malicious AODV: Implementation and Analysis of Routing Attacks in MANETs. In: Procedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2012), UK, pp. 1181–1187 (June 2012)
3. Lima, M., dos Santos, A., Pujolle, G.: A survey of survivability in mobile ad hoc networks. IEEE Communications Surveys & Tutorials 11, 66–77 (2009)
4. Dengiz, O., Konak, A., Smith, A.E.: Connectivity management in mobile ad hoc networks using particle swarm optimization. Ad Hoc Networks 9, 1312–1326 (2011)
5. Cho, Y., Smith, J.S., Smith, A.E.: Optimizing tactical military MANETs with a specialized PSO. In: 2010 IEEE Congress on Evolutionary Computation (CEC), pp. 1–6. IEEE (2010)
6. Miles, J., Kamath, G., Muknahallipatna, S., Stefanovic, M., Kubichek, R.F.: Optimal trajectory determination of a single moving beacon for efficient localization in a mobile ad-hoc network. Ad Hoc Networks (2012)
7. Asadinia, S., Rafsanjani, M.K., Saeid, A.B.: A novel routing algorithm based-on ant colony in Mobile Ad hoc Networks. In: 2010 3rd IEEE International Conference on Ubi-media Computing (U-Media), pp. 77–82. IEEE (2010)
8. Daniel, A.K., Singh, R.: Swarm Intelligence Based multicast Routing and Bandwidth Management protocol for Ad-hoc wireless Network Using Backpressure Restoration. In: 2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), vol. 5, pp. 516–520. IEEE (2010)
9. Fahmy, I.M.A., Hefny, H.A., Nassef, L.: PEEBR: Predictive Energy Efficient Bee Routing algorithm for Ad-hoc wireless mobile networks. In: 2012 8th International Conference on Informatics and Systems (INFOS), pp. NW-18–NW-24. IEEE (2012)
10. Taggu, A., Taggu, A.: An application-layer scheme for intrusion detection in MANET using mobile agents. In: 2011 Third International Conference on Communication Systems and Networks (COMSNETS), pp. 1–4. IEEE (2011)
11. Roy, D.B., Chaki, R.: Detection of Denial of Service Attack Due to Selfish Node in MANET by Mobile Agent. In: Özcan, A., Zizka, J., Nagamalai, D. (eds.) WiMo/CoNeCo 2011. CCIS, vol. 162, pp. 14–23. Springer, Heidelberg (2011)
12. Mohamed, Y.A., Abdullah, A.B.: Immune-inspired framework for securing hybrid MANET. In: IEEE Symposium on Industrial Electronics Applications, ISIEA 2009, vol. 1, pp. 301–306. IEEE (2009)
13. Ye, X., Li, J.: A security architecture based on immune agents for MANET. In: International Conference on Wireless Communication and Sensor Computing, ICWCSC 2010, pp. 1–5. IEEE (2010)
14. Halim, I.T.A., Fahmy, H.M.A., Bahaa El-Din, A.M., El-Shafey, M.H.: Agent-Based Trusted On-Demand Routing Protocol for Mobile Ad Hoc Networks. In: 2010 4th International Conference on Network and System Security (NSS), pp. 255–262. IEEE (2010)
15. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, pp. 39–43 (1995)
16. Sánchez-Casado, L., Maciá-Fernández, G., García-Teodoro, P.: An Efficient Cross-Layer Approach for Malicious Packet Dropping Detection in MANETs. In: Procedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2012), UK, pp. 231–238 (June 2012)

Modeling Dependence Networks for Agent Based Simulation of Online and Offline Communities

Francesca Marzo, Stefano Za, and Paolo Spagnoletti

CeRSI – LUISS Guido Carli University, Rome, Italy
{fmarzo,sza,pspagnoletti}@luiss.it

Abstract. Far from simply being a concept useful in investigating social relationships, social networks are rapidly becoming a diffuse phenomenon to deal with in everyday life. The goal of this paper is to provide insights from the design research perspective, both for online and offline communities. Starting from the idea that the phenomenon under investigation emerges from the interaction of autonomous agents in an environment in which other agents interact with each other in order to reach their own goals, we adopt a Multi-Agent Simulation (MAS) approach to study social networks dynamics of online and offline communities. In particular, we built an agent-based simulation of dependence networks, considered crucial for the interaction of cognitive agents and for the exchange of resources between them. As results we have been able both to better define some hypotheses on dependence networks dynamics and to highlight possible future research particularly useful for the design of digital platforms.

Keywords: Social networks, Communities, Dependence Networks, Cognitive Agents, Multi-Agent Simulation.

1 Introduction

With the rapid growth of online social networking and open source models, individuals are increasingly engaged in online activities by interacting with peers and organizations through digital platforms and applications running on personal computers and mobile devices [1–3]. Such social network applications have the potential to facilitate the exchange of resources among members by providing platforms for the exchange of ideas and quality information [4]. When effectively designed and implemented, digital platforms enable information exchange, collaboration and collective action within online and offline communities [5]. Although the behavior and motivation of members participating in online (virtual) communities has been widely investigated, more research is needed on how to design, build, and sustain a vibrant platform [2].

The presence of multi-level factors in the ecological model of community behavior, poses many challenges when investigating the mutual relationship between agency and structure in this context [6]. Previous studies on social networks mainly focus on a behavioral approach that reflects a positivist stance by performing both quantitative and qualitative analysis for validating causal models. Our goal is to

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 192–203, 2013.

complement these studies by adopting a constructivist viewpoint which extends the methodological framework for investigating this phenomenon and provide insights also from a design research perspective [7]. Drawing on complex adaptive systems theory, we recognize that social phenomena emerge from the bottom-up interactions among learning agents in a given environment [8]. This view has recently gained much attention in different fields of management and organizations [9–12] and integrate contributions from cybernetics, cognitive sciences, decision and organization sciences [13].

In this paper we present an agent based model for simulating the dynamics of small groups of interacting agents whose behavior is determined by their learning capability and a set of environmental rules. The model is grounded on the theory of dependence networks [14, 15] and provides a tool for studying emergent properties/phenomena within online and offline communities. Following previous works that introduce formal models of dependence networks, we develop an architecture of cognitive agents and of the environment in which they act and interact. This architecture will be the basis for implementing a platform for agent based simulation that will serve as a tool for supporting decision making in the design of digital platforms and in the governance of online and offline communities.

The paper is structured as follows. We first introduce the theoretical framework on which the model is grounded. Then we describe the architecture of the model and we illustrate an example of simulation. Finally we discuss about implications and research directions.

2 Cognitive Model of Dependence Networks

Starting from Conte and Castelfranchi's theory [16] of dependence network, we assume the idea that agents are part of a network of social relationships and that these relationships are fundamentally based on agent's mental states. Therefore, social networks are built on networks of goals. Among these relationships, one of the most important type is social dependence, consisting in some agents needing other agents' to reach their goals.

We take our move from a previous study on agent-based simulation of dependence network aimed to supply a tool for improving coordination in multi-agent systems [17] and we integrate this model by assuming an epistemically different point of view: there exists an objective reality not necessarily known by agents as it is. Indeed, with reference to Conte and Sichman formal theory of social dependence [17], we propose an updated model of dependence networks based on the evolution described in more recent works [14]. Therefore we extend the three basic notions of the first model, i.e. external description, dependence relationship and dependence situation. In particular, while we build external descriptions and dependence relationships following the instruction presented in the previous model of dependence network we consider some additional features of dependence situations in order to classify them. Together with the nature (given by dependence relations) and the locality (considered by Conte and Sichman), we add the distance between the locally believed dependence and the real dependence. In other terms, we extend the model by explicitly considering subjective and objective points of view in order to test how their distance influences agents' behaviors in the networks. Therefore, the cognitive model we used as theoretical framework takes into account both an objective dependence network, built on the real dependence relation between agents in the network, and multiple believed dependence networks (as many as the number of agents in the network).

We can consider the model as made up by an environment exogenously given, characterized by different dependence relations, and a set of goal oriented agents autonomous in making decision but dependent by other agents to reach their own goal. On the base of their locally believed dependence network, agents can proceed by trial and error in order to reach the goal (updating their own beliefs about the network in a process that can lead to reduce the divergence between it and the real environment).

As for this first attempt, in order to make it as simple as possible, we tested only a single type of dependence relationships among the set defined in [17], namely the mutual one, which means that agents depend on some of the others in the network to reach a common goal. The idea is to test interactions more and more complex once the simulator described in the next section is built and operative.

3 The Architecture for Simulation

Each agent in the environment has her own representation of the reality on the basis of her beliefs. She well knows her available actions and resources, and she starts to reach some goals, combining her actions and resources and/or asking some of them to other agents in the environment. In this way, she interacts with other agents either to perform an exchange of resources or to involve other agents in performing a specific action. The interaction is based on the believed dependence network, and it can be near or not to the reality. After every interaction the agent can update her believed dependence network on the basis of the information exchanged. The interaction among the agent happens every slot of time defined as rule of the environment (called round). For example: if an agent needs a resource or an action but she doesn't know who can perform that action or give her that resource, she sends the request to the environment (as a broadcast message). In this case every agent in the environment receives that request and can contact directly the requester or broadcast the answer (on the basis of the simulation design). In this manner all agents know what the applicant agent needs. At the same time the latter can discover a new dependence link with an agent (not necessarily a new link in general) from which she depends for a resource or an action. Furthermore everybody in the network, if the answer was broadcasted, may discover new information (new agent and/or new dependence link). In the end, this kind of information is stored in the work memory and afterwards it will be used to update the believed dependence network. We can therefore summarize as follows the research questions that guide our simulation: [RQ1] Does a knowledge of dependence network as much as possible close to the real configuration give an advantage to the agent that has that knowledge? In other terms can that agent reach the goal before the others? [RQ2] Is being part of a network where information is broadcasted an asset for agents in the network?

3.1 Agent Mind and Environment Configuration

Each agent has two kind of memories for storing several information, namely the Long Term Memory (LTM) and the Working Memory (WM). The former contains all information needed by the agent to act (goals, actions, plans, resources, etc.), whereas the latter is used to know what an agent will do in the next round and also to store the information produced by the interactions in the environment. This last kind of information may also be useful to update in the LTM the Believed Dependence Network by the agent. In particular the Long Term Memory (LTM) contains:

- a set of goals $G = \{g_1,\ldots,g_n\}$;
- a set of plans $P = \{p_1,\ldots,p_n\}$ where each p_i is the collection of actions to reach the goal g_i by the agent (each plan can be updated in runtime if needed);
- a set of possible actions $Act = \{a_1,\ldots,a_m\}$ where for each action one or more resources are associated
- a set of resources $R = \{r_1,\ldots,r_l\}$ owned by the agent
- a network dependence of the society built on the basis of the agent's beliefs
- a set of rules (Rules for Updating - RU) with which the information present in the Working Memory can be considered reliable and useful for updating the believed dependence network and the RU themselves.

As regards the Working Memory (WM), it contains the step of the plan to execute, and the stored information received (obtained during the interaction). The acting of agents is driven not only by the information stored in their own LTM but also by the information or constraints inherited from the environment. The environment settings contains:

- the real dependence network (it can be unknown to all the agents)
- the set of priority rules for executing some actions (not necessarily they involve all possible actions (i.e. some actions are executable independently from others)
- the set of possible resources needed for executing an action (i.e. some actions require using certain resources)
- the information about the latency between one round and the next one (an arbitrary technical requirement).

3.2 A Simple Scenario

As a first step of our research, we consider a simplified and generic scenario with the following assumptions:

- The Goal is the same for each agent: consuming resources following a given sequence; each agent can start from a different position in this sequence depending on the resources she has; she reaches the goal if the sequence is complete. Given the goal G={R1,R2,R3,R4} if the agent Ai has the set of resources R={R3,R5,R3,R2,R4}, she will start to find a combination of her resources as to reach the longest sequence she can: R2, R3, and R4; then she must start looking for

R1; once obtained R1 by another agent, she reaches the goal G (in this example in five rounds should she have received R1 in one interaction); resources not needed to achieve G, can be externally provided upon request (i.e. R3 and R5 in the example)

- The actions for each agent can be: Act = {consume a resource; ask for a resource to a given agent; ask for a resource through a broadcasting request}; for now we consider the only possibility for an agent to give the requested resource, when that resource is not needed by the owner (in which case it will be pre-allocated to be consumed by the latter in next rounds)
- Each agent can execute only one action in a given round
- The number of resources owned by the agents either is the same or it is higher than the number of resources for achieving the goal. In this scenario resources are generic items owned by the agents, in order to allow further specification in different application domains
- The unique set of given priority rules is related to the sequence of the resources to consume
- Each agent has her own believed dependence network (on which she bases her interactions)
- There is a unique updating rule: the new information collected in a certain round updates the believed dependence network and it will be used in the next round.

4 Simulation Design

According with the rules and assumptions described in the previous paragraph concerning the definition of a simple scenario, we describe here the simulation settings.

4.1 Environmental Settings

In the simulation each agent has the common goal G associated with a consume of resources that follows the sequence {①②③④⑤⑥⑦} in a circular way. This means that in case an agent starts in the middle, the previous sub-sequence is shifted to the end. As pointed out in previous section, possible actions of each agent are: allocate the resources (ALL), consume a resource (CON), ask for a resource to a given agent (ARA), ask for a resource as a broadcasting request (ARB), broadcast an answer (BRA), wait for the BRA. Moreover for each round each agent must perform only one of the following actions:

- ALL: in the first round all the agents strategically allocate their resources in order to reach their goal; the resources not allocated are made available for answering possible requests; in the next rounds each agent consumes the allocated resources starting from the resource that allows her to consume the longest sequence without asking for missing resources;
- CON: in each round an agent can consume one of the allocated resources or one received in the previous round, following the sequence imposed by the goal ($A_i(CON(R_j))$=agent Ai consumes resource R_j);

- ARA: if an agent needs a resource and knows who can give it to her then she will ask directly ($A_i(ARA(R_j,A_k)$= agent A_i directly ask for R_j to A_k); if A_k owns the requested resource A_i receives and consumes it in the same round by notifying the transaction to all the agents; if A_k does not have R_j, A_i has wasted a round;
- ARB: if an agent needs a resource and does not know who can give it to her, then she will broadcast the request ($A_i(ARB(R_j)$ = agent A_i broadcast the request for R_j);
- BRA: when agents receive a broadcasted request they must answer with a broadcast message only in case they have that resource available ($A_i(BRA(R_j)$= agent A_i broadcasts her answer about the availability of R_j).
- WAIT: A_i(WAIT)= agent A_i wait for the answer

In this first simulation, note that if a resource is requested by multiple-agents, the agent who gets the resource will be randomly selected. This policy will be changed in future works, by adding variables able to influence the allocation of requested resources (such as level of trust, level of dependence etc.). Also the agent schedule for activation (i.e. who acts first) is random for this simple scenario but will be subject to modification in the future.

For the simulation run we illustrate, agents can update their BDN following the common rule that the new information are used at the end of each round for updating the believed dependence network. In future works, in order to understand some important phenomena such as how results are influenced by ordering agents' activation, we will modify the policy of random choice and we will adapt it to specific contexts. Furthermore, to give more straight to the model, we will test the implication of dealing with a bigger number of agents.

The initial condition is that there are 4 agents: A1, A2, A3, and A4 with the following sets of resources distributed randomly with the additional condition that the totality of resources is sufficient for making each agent reach her goal:

- R_{A1}={③③④④⑤⑥⑦}
- R_{A2}={②②③③⑤⑤⑥}
- R_{A3}={①①④④⑤⑥⑦}
- R_{A4}={①①②②⑥⑦⑦}

Fig. 1 shows the Real Dependence Network Graphs (RDN graphs), where each node represents the agent whereas ties and their labels represent the dependence links between agents concerning a specific set of resources. Considering the classification proposed in [17], in this scenario only a Mutual Dependence is allowed, because each agent has a potential dependence tie with everyone in the environment for the same goal, as shown in the figure below.

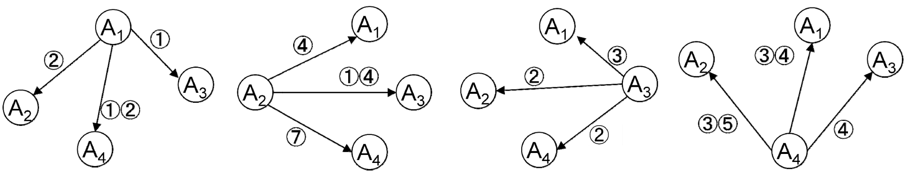


Fig. 1. Real Dependence Network Graphs

Starting from the RDN graphs we build an adjacency matrix named Real Dependence Network Matrix (RDN matrix) in which each row is related to one specific agent and contains the resources upon which the agent depends on the others. Since all ties show a mutual dependence, every cell (except those in the diagonal) contains at least one resource, as show in the table below.

Table 1. Real Dependence Network matrix

Really Needed	A1	A2	A3	A4
A1		②	①	①②
A2	④		①④	⑦
A3	③	②		②
A4	③④	③⑤	④	

The use of matrix for describing the dependence network is useful to represent the believed dependence network (BDN) for each agent acting in the environment, as show in the **Table 2**.

4.2 Agent Settings

The behaviour of an agent can be described as follows. The agent starts with the allocation of the resources (ALL), sorting them and defining the longest sequence from which to start for consuming resources. Afterwards, until the goal is achieved, she performs one of the following actions on the basis of her state. Either she answers to a broadcast request, or she proceeds towards her goal. In this latter case either she consumes one of the owned resources, or she asks for the resource to another agent assuming that the latter has that resource available. When an agent does not have the resource and does not to which agent to ask, she performs a broadcast request to all agents in the environment (this is the only action that requires a further round for waiting the broadcast answer). Therefore the agent's behaviour is led by the resource availability and by the believed dependence network.

In our scenario the four agents have a different perception of the environment. They do not know exactly their real dependence links with the other agents: they have not a complete vision of the dependence links and sometimes they have wrong assumptions. Following we resume the main aspects emerging from the observation of the dependence matrix for each agent in **Table 2**:

- agent A1 knows all the agents but she ignores one dependence link with A3 about the resource ①; furthermore she supposes wrong links about the five resources underlined in **Table 2.**
- agent A2 ignores the presence of agent A1 in the environment – grey column – and hence she ignores also the dependence link with A1 and her dependence link with A3 about resource ④;
- agent A3 is the unique agent that has a complete and exact vision of the dependence network (BDN is equal to RDN);

- agent A4 ignores the presence of agent A3 in the environment – grey column – and hence she ignores also the dependence link with A3 and her dependence link with A1 about resource ④.

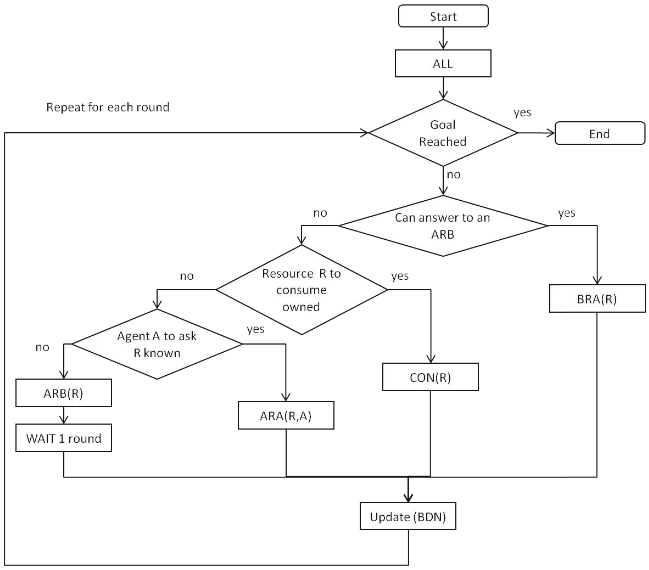


Fig. 2. Agent choice of action flow diagram

Table 2. The Believed Dependence Network Matrix for each agent in the environment

Really Needed	$DD(A1)=12\ BDN_{A1}$				$DD(A2)=6\ BDN_{A2}$				$DD(A3)=0\ BDN_{A3}$				$DD(A4)=8\ BDN_{A4}$			
	A1	A2	A3	A4	A1	A2	A3	A4	A1	A2	A3	A4	A1	A2	A3	A4
A1		②	④	① ②		②	①	① ②		②	①	① ②		②		①
A2	③		④	⑦			①	⑦	④		①④	⑦				
A3	④	②		②		②		②	③	②		②	③	②		②
A4	③ ④	④	⑤			③ ⑤			③④	③ ⑤	④		③	③ ⑤		

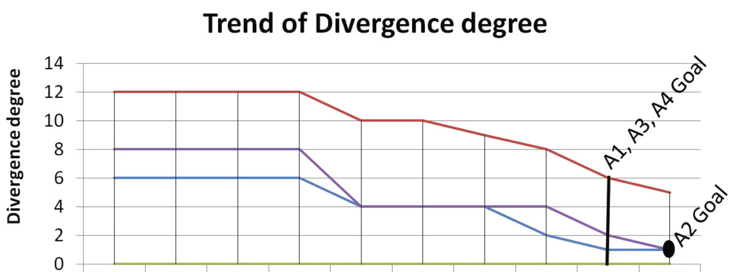
With this setting, during the simulation agent A2 or A4 may ask for the resource ④ because at the start time they do not know who can provide it in the environment, unless they acquire that information from previous interactions performed by others. In fact, as mentioned in the **Fig. 2** at the end of each round the agent can update her BDN on the basis of information arisen during the interaction among all the agents (respecting the unique updating rule RU).

4.3 Simulation Run

Taking into account the research questions mentioned in section 3, and the scenario described before, we formulate the following hypotheses to be tested in the simulation run:

- H1: A3 reaches the goal before the others;
- H2: A2 or A4 can update their knowledge through broadcasted answers, to the request "who has the resource ④?"; therefore either A2 or A4 can take advantage (one round) from the information circulated following a broadcast request from A4 or A2 respectively;
- H3: Round after round all the BDNs converge to the Real Dependence Network.

H1 and H2 concern advantages at the individual level, whereas H3 is related to a collective improvement (network performance). The figure below traces the results of the simulation until each agent reaches her goal. Round by round, the distance between an agent believed dependence network and the real dependence network, named Divergence Degree (DD), is represented. The DD is calculated as the sum of the missed links and the wrong links for each resource (i.e. if A1 ignores her dependence from A2 about the resources R5 and R6, the DD is equal to 2). An agent reaches the goal when she has consumed the sequence of resources (also shifted compared to the original one).



Rounds	1	2	3	4	5	6	7	8	9	10
Agent A1	12	12	12	12	10	10	9	8	6	5
Agent A2	6	6	6	6	4	4	4	2	1	1
Agent A3	0	0	0	0	0	0	0	0	0	0
Agent A4	8	8	8	8	4	4	4	4	2	1

Fig. 3. Simulation run

As depicted in **Fig. 3**, the simulation shows the convergence of the BDN to the real dependence network for all the agents, as supposed in H3. When all the agents have achieved their goals DD is equal to 5 (4 wrong links) for A1, 1 for A2 and A4, and 0 for A3. The decreasing of DD starts decreasing after the fourth round, when some agent start sending messages for retrieving resources. The most relevant rounds, in which a variation of DD takes place, are described below. In each round "DD(Ai)" indicates the DD for the agent "Ai".

- Round 5: agents A1 and A3 answer in broadcast to the A2's request for ④ made in the round 4; A2 and A4 discover their dependence link with A1 and A3, but A2 knows only her two links; A4 knows both her two links and also those of A2; therefore DD(A4) and DD(A2) decrease of 4 and 2 units respectively; furthermore A1 discovers that A2 does not have the resource ④, and that the latter depends on the former, so DD(A1) decreases of 2 units;
- Round 7: agent A4 takes ③ from A2; A1 discovers a dependence link between A4 and A2 gaining 1 unit for her DD;
- Round 8: among the three ARA actions performed in this round only the action of A4 (A4(ARA(④, A3))) produces an upgrade of some BDNs; in fact the DD(A1) and DD(A2) decrease of 1 and 2 units respectively;
- Round 9: agent A1 takes ② and agent A2 takes ⑦ from A4, implying a decrease of DD(A4) of 2 units; agent A3 takes ③ from A1 and A4 takes ⑤ from A2, producing a reduction of DD(A1) for 2 and DD(A2) for 1 units;
- Round 10: only A2 still need to achieve the goal and she takes ① from A3; this action allows to A1 and A4 to decrease their DD for another 1 units.

Considering the rounds described so far, H3 is confirmed, in fact BDN converge to RDN in all cases. Also H2 is confirmed, since the agent A4 benefits from the request asked by A2 (in the fourth round). Actually the agent A4 gains more than A2, in fact in that round she gains 4 units of DD while A2 only 2. The only hypothesis that is not confirmed is H1. In fact the agent A3, which knows from the initial state the overall RDN, does not reach the goal before all the other agents but at the same time as A1 and A4; only the agent A2 (the only agent that sent a broadcast request) achieves the goal one round after A3.

5 Discussion and Conclusion

The simulation run, performed as a pilot for the computer based simulation that is under development, allows to highlight some interesting dynamics of the dependence networks. Considering both the objective and the subjective point of views, such dynamics are characterized in different ways. In the RDN there are only updates related with the exchange of resources. In fact when an exchange is performed the related dependence tie disappears, producing an information update. In the BDN, dynamics are mostly characterized by the information exchanged when agents ask for resources by the means of an intentional communicative act (i.e. broadcast request).

Although in both cases the exchange of resources produces information, in the BDN a communicative level provides an additional means for learning and for achieving goals. In fact in RDN and BDN the exchange of resources implies, at the information level, an environmental update and a cognitive update respectively. Furthermore broadcast communication (requests and responses) allows agents to update their BDN and to reduce the gap with RDN. As consequence it represents an advantage to achieve the goal for agents in the network. This can be considered as a first positive answer to RQ2, to be deeply investigated in a more complex scenario.

Another aspect concerns the relationship between the cognitive representation of the real environmental configuration and the performance of the agent. The extent to which RDN and BDN overlap (DD) is intuitively related to the capabilities to reach the goal: the more they overlap the faster an agent achieve her goal. However the simulation shows that this behaviour does not holds in such environment: pilot results show that the advantage derived from having DD equal to 0 can be caught up by other agents thanks to the communicative level (e.g. broadcasting). Nevertheless additional evidences are needed to further investigate this phenomenon, and hence fully answer RQ1. A possibility is to consider more complex scenarios in which, for example, agents can deceive or ignore the requests.

By answering the two research questions, this study provides interesting insights on the behaviour of communities, both online and offline, in which members interact for achieving their personal goals. First, under the environmental settings modelled in our simulation, a complete knowledge of the real dependence network does not represent a sufficient condition for allowing members of the network to reach their goals in a shorter time frame with respect to other members whose perception of the resource distribution is far from the real situation. Second, being part of a network in which it is possible either to broadcast messages and to receive requests from the environment allows members of a community to exchange resources purposefully and to achieve their individual goals.

Clearly, since this is only a first attempt to use agent-based simulation for the domain under investigation, several improvements in next steps are already considered as necessary. Among them, as already pointed out, we are considering to test our model with a bigger number of agents involved, so that online and offline communities specific features can be taken into account. In fact, it can be hypothesized that the topology of the network will have great impact when considering larger networks. As well as the size and the type of networks, we will study the importance of different levels of dependence and different dynamics that can characterize different networks and communities (i.e. how agents enter and exit from the dependence networks can influence how quickly BDN and RDN converge).

These results have important implications for guiding the emergence of desired outcomes in the governance of online and offline communities. In fact the research shows the potential of adopting agent based models, which mix objectives and subjective views of the reality, for exploring the effects of structural characteristics of networks (environment) and of agents' cognitive models. Additional insights can be gained by representing other aspects of cognitive models and environmental constraints such as for instance trust and digital platforms functionalities (i.e. public and private groups, noticeboard, feeds, etc.). Furthermore, computational simulations can provide a means for exploring multiple trajectories of community behaviours in some specific domain by taking into account the complex nature of the phenomena under investigation and hence complementing with a constructivist approach the mainstream of positivistic studies of social networks. Therefore we argue that future simulation studies in management and information systems [18] can benefit from the proposed architecture in the design and evaluation of digital platforms and their governance models.

References

1. Spagnoletti, P., Federici, T.: Exploring the Interplay Between FLOSS Adoption and Organizational Innovation. Communications of the Association for Information Systems 29, 279–298 (2011)
2. Yoo, Y., Boland Jr., R.J., Lyytinen, K., Majchrzak, A.: Organizing for Innovation in the Digitized World 23, 1398–1408 (2012)
3. Resca, A., Spagnoletti, P., Za, S.: An attempt to outline strategic frameworks facing digital innovation. In: Innovation in Information Infrastructures (III 2012), Edinburgh, UK, October 9-11 (2012)
4. Shirky, C.: Here Comes Everybody: The Power of Organizing Without Organizations. Penguin Press, New York (2008)
5. Spagnoletti, P., Resca, A.: A Design Theory for IT supporting Online Communities. In: Proceedings of the 45th Hawaii International Conference on System Sciences, pp. 4082–4091 (2012)
6. Resca, A., Tozzi, M.: Offline and Online Communities: Great Differences and Some Similarities. In: Baskerville, R., De Marco, M., Spagnoletti, P. (eds.) Designing Organizational Systems. LNISO, vol. 1, pp. 310–318. Springer, Heidelberg (2013)
7. Spagnoletti, P., Baskerville, R., De Marco, M.: The Contributions of Alessandro D'Atri to Organization and Information Systems Studies. In: Baskerville, R., De Marco, M., Spagnoletti, P. (eds.) Designing Organizational Systems. LNISO, vol. 1, pp. 1–18. Springer, Heidelberg (2013)
8. Holland, J.: Emergence: From chaos to order. Basic Books, New York (1998)
9. Lewin, A.Y.: Application of Complexity Theory to Organization Science. Organization Science 10, 215 (1999)
10. Allen, P.M., Varga, L.: A co-evolutionary complex systems perspective on information systems. Journal of Information Technology 21, 229–238 (2006)
11. Amaral, L.A.N., Uzzi, B.: Complex Systems–A New Paradigm for the Integrative Study of Management, Physical, and Technological Systems. Management Science 53, 1033–1035 (2007)
12. Anderson, P.: Complexity Theory and Organization Science 10, 216–232 (2008)
13. Simon, H.: The Sciences of the Artificial. MIT Press (1996)
14. Castelfranchi, C., Falcone, R., Marzo, F.: Being Trusted in a Social Network: Trust as Relational Capital. In: Stølen, K., Winsborough, W.H., Martinelli, F., Massacci, F. (eds.) iTrust 2006. LNCS, vol. 3986, pp. 19–32. Springer, Heidelberg (2006)
15. Marzo, F., Castelfranchi, C.: Trust as individual asset in a network: A cognitive analysis. In: Spagnoletti, P. (ed.) Organization Change and Information Systems. LNISO, vol. 2. Springer, Heidelberg (2013)
16. Conte, R., Castelfranchi, C.: Simulating multi-agent interdependencies. A two-way approach to the micro-macro. In: Mueller, U., Troitzsch, K. (eds.) Microsimulation and the Social Science. Lecture Notes in Economics. Springer, Heidelberg (1996)
17. Conte, R., Sichman, J.S.: DEPNET: How to Benefit from Social Dependence. Journal of Mathematical Sociology 20, 161–177 (1995)
18. Za, S., Spagnoletti, P.: Knowledge creation processes in Information Systems and Management: Lessons from simulation studies. In: Spagnoletti, P. (ed.) Organization Change and Information Systems. LNISO, vol. 2. Springer, Heidelberg (2013)

Improving Classifier Agents with Order Book Information

Philippe Mathieu and Matthis Gaciarz

Laboratoire d'Informatique Fondamentale de Lille
LIFL UMR 8022 CNRS
Lille1 University
59655 Villeneuve d'Ascq, France
philippe.mathieu@lifl.fr

Abstract. In the study of financial phenomena, agent-based artificial markets are efficient tools for testing economic assumptions of market regulation. While it is easy to populate these virtual worlds with "basic" chartist agents using price history (increase or decrease of prices, moving averages...), it is nevertheless necessary, in order to study rationality phenomena and influence between agents, to add some kind of learning agents. Several authors have of course already been interested in adaptive techniques but they mainly take into account price history. But prices are only consequences of orders and therefore reasoning about orders provides a head start in the deductive process. In this paper we show how to take into account all of the information about the market, including how to leverage the information from order books such as the best limits, size of bid-ask spread or cash at hand waiting to accommodate more effectively to market offerings. Like B. Arthur we focus here on the use of LCS agents.

1 Introduction

For a few years, advances in computer science have provided powerful tools to study complex economic systems. Individual-based approaches provide a high level of detail and various advantages. It is now possible to test regulation systems, or the influence of new policies on an individual scale, and not only group scale. Among these economic systems, artificial stock markets now offer a credible alternative to mathematical finance and financial econometrics. With these tools, macroscopic phenomena become the consequence of microscopic phenomena, because decision processes and actions are made by the individuals. Thanks to these advantages, multi-agent systems show all their potential.

1.1 A Multi-agent Artificial Stock Market

In the literature many artificial markets use an equation based price fixing mechanism [1] [2]. In these markets, the agents send bid or ask signals on the market and the market fixes prices according to the total number of each signal. This

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 204–215, 2013.

method is different from how real marketplaces work. Moreover it is not powerful enough to allow endowing agents with powerful intelligent strategies, reasoning on the type, the price and the quantity of each order. This level of detail is required to test the consequences of regulation rules on individuals, the social well-being of societies [3] or the influence of speculators in the trader population.

The artificial market platform ATOM [4], is a tool that clones the main features of the Euronext-NYSE stock exchange microstructure. One of these features is the double-auction order-book system : for each asset, the ask part contains orders with prices in increasing order and the bid part contains orders with prices decreasing order. Like Euronext, ATOM aims at matching orders sent by virtual traders to determine quotations and prices. Orders are sent by agents that all have their own strategy.

With its realistic design, ATOM allows agents to access to a lot of data. There is the price history (Fig. 1), that is the only information used by equational systems, but also the order history and the order book (Fig. 2). The order book shows the state of the bid and the ask at time t, and contains information about the orders. Thus ATOM allows agents to take advantage of much information to develop their strategies.

Price	...	110.8	110.9	111.0	110.9
Quantity	...	5	2	7	8

Fig. 1. History of fixed prices, with the traded quantities

Dir	Order	Issuer	Quantity	Price
	...	...	...	...
	o1	Ag2	8	111.5
	o2	Ag1	10	111.1
Ask	o4	Ag3	3	**111.0**
			bid-ask spread ↕	
Bid	o10	Ag4	2	**110.8**
	o3	Ag1	6	110.6
	...	...	...	...

Fig. 2. Order book for one particular asset. The best prices ar shown in bold font. The interval between the best offers is called bid-ask spread.

For many years, economic theories have asserted that it is important to take the information of the order books into account [5] [6] [7], but no one has highlighted it experimentally so far until now.

The goal of this paper is to show that it is possible to design agent behaviours that take advantage of this information to build more rational behaviours.

1.2 Learning Agents

Like multi-agent systems, machine learning techniques have been developed over the last decades. The main learning categories (supervised, unsupervised, reinforcement learning) are based on various algorithms [8].

Many of them have been used on artificial markets, with more or less success [9] [10]. Nevertheless, all these techniques have a lack of explanation: once the learning has been achieved, it is difficult to understand why the decisions are taken, or why specific behaviours have been triggered.

Like B Arthur, we choose to support this work with a learning technique not often used in literature, but more suitable for explanatory models: classifier systems. Other techniques, like the ones we mentioned above, would work too. The purpose of our work is not to compare these, but to show how to design an adaptive and explanatory model that uses all the market information to provide sophisticated and varied behaviours. A Learning Classifier Systems (LCS, [11]) contains a set of binary rules. The quality of a rule, called fitness, is modified at each step by a reinforcement learning technique. Moreover, a genetic algorithm periodically performs a natural selection, based on classical mutation and selection mechanisms.

One of the first artificial markets, the SF-ASM (Santa Fe Artificial Stock Market, [1] [2]), already used LCS for the agent reasoning, but it was an equational market, and its agents can only take the price history in account.

1.3 Classifier Systems

Before tackling the whole complexity of a market, let us describe how a usual classifier system works [1], illustrating it within the framework of an equational price-fixing market, using agents that only decide to buy or sell.

An LCS uses a set of conditions on market state called market descriptors. These descriptors can be seen as the "sensors" used by the LCS to perceive the market. With these, it is possible to analyse the market and build a binary sequence whose length is equal to the number of descriptors used. Figure 3 shows an example of a descriptor set. The first descriptor Ex1 is satisfied if the current price is higher than the previous price. Ex2 is satisfied if the current price is higher than the average of the last five prices. Ex3 is satisfied if the current price is less than 100.

Once these descriptors have been chosen, each LCS is endowed with set of rules (or classifiers) from a triplet (state, score, action):

- State S of a rule is a trinary sequence that determines whether this rule can be activated considering the current situation. The trits (**tr**inary dig**its**) can be set to 1, 0 or #. In the sequence, each trit matches a descriptor. If the value of a trit is 1, the descriptor has to be satisfied to activate the rule. If

Ex1	$p_t > p_{t-1}$
Ex2	$p_t > 1/5 \times \sum_{i=t-1}^{t-5} p_i$
Ex3	$p_t < 100$

Fig. 3. Descriptor examples that can be used by a LCS

it is 0, the descriptor has to be unsatisfied to activate the rule. The sign # is a wildcard, meaning that the descriptor is not taken into account in the rule's activation process.
- the fitness score F represents the trust allotted to this rule, according to its previous forecast successes
- the action A represents the action to perform when the rule is activated

An LCS can have up to 3^n rules to reason on, n being the number of descriptors. Of course, it can have fewer.

Figure 4 is an example of a five-rule classifier system using three descriptors. for instance, the first rule R1 can be activated if the current situation of the market matches with state 010 or with state 110. This rule allows the LCS to select a bid order, and its score is 5.

	state	action	score
R1	#10	buy	5
R2	1#0	sell	18
R3	00#	buy	12
R4	110	buy	4
R5	#11	sell	9

Fig. 4. Example of a 5 size classifier system

An LCS works as follows : each time it has to take a decision, it selects a rule among the activable ones, with a random mechanism in which the probability for each rule to be chosen is proportional to its score.

The LCS agent performs the action of the selected rule (it sends a bid or a ask signal). During the next LCS activation, the score of each activable rule is updated. It is increased or decreased according to the correctness of the forecast produced by the rule.

A genetic algorithm is periodically applied on the set of rules, using the score of each rule to perform a natural selection in the classifier system. If a score does not reach a threshold, the rule is eliminated. The best rules are crossed and mutated to regenerate the population. Combining the continuous evaluation of the rules and the genetic algorithm allows the LCS to perform an effective learning routine. It is possible that some rules cannot be activated because of contradictory descriptors. To avoid this and keep the classifier system size constant, the rules not activated for a long time are eliminated by the genetic algorithm.

2 Proposal

Setting up LCS in each agent in an order-driven market raises several problems. The previous LCS only chooses directions, but in an order-driven market, other information is necessary for an agent to build an order. The most common being the LimitOrder in which the agent has to generate a quantity and a price.

Moreover, it is important to take advantage of pending orders. Finally, setting up an LCS requires to clearly define temporal references for the descriptors that can be interpreted in various ways.

When an agent sends an order to the market, the direction of this order is determined by the activated rule. On a real market, and on ATOM, it is also necessary to set a price limit and a quantity for this order. To do this, we propose to use a simple policy, made up of a price-setting strategy and a quantity-setting strategy. For each of these two strategies, there are various possibilities:

Price-setting strategies :

- Setting the price so that the order sent is placed at the top of the order book:
  $P_{Bid} = P_{BestBidOrder} + \varepsilon$
  $P_{Ask} = P_{BestAskOrder} - \varepsilon$
- Producting an order that will be immediately executed (at least partially), setting a price equal to the best rival order's price :
  $P_{Bid} = P_{BestAskOrder}$
  $P_{Ask} = P_{BestBidOrder}$

To set the quantity, various strategies are possible too.

Quantity setting strategies:

- Constant quantity : $Q = k_c$
- Quantity proportional to the chosen rule's score : $Q = k_p F$

Our experiments have shown that the policy which give the best result is the policy that sends orders to the top of the order book, with a constant quantity. The next results will be based on this policy.

In an asynchronous order-driven market, other information is available for the agent, based on the past orders and the current order book. We propose to use this information in the classifier system.

2.1 Improve Agents Behaviour Learning from Orders

As equational systems are the most widespread ones, the common reasoning in finance is to perform a technical analysis of the price history to deduce a future increase or decrease. It is the case of "chartist" agents, that search for particular shapes in the price series. In order to show the contribution of the order book to agent reasoning, in this paper we propose to use a price-based LCS agent and to improve it adding new order-based descriptors. Then with a set of experiments we show that these descriptors provide relevant and useful information for decision-making.

The agent used as a reference in this work uses the price-based criteria introduced in Fig. 5. These are interesting because they allow to take into account price variations both in the short and the long term, on three simple criteria : price increase compared to an older price (descriptor 1), compared to the average of the last n prices (descriptors 2 to 4), or compared to the midrange of the last n prices (descriptors 5 and 6).

1	$p_t > p_{t-1}$
2	$p_t > 1/5 \times \sum_{i=t-1}^{t-5} p_i$
3	$p_t > 1/10 \times \sum_{i=t-1}^{t-10} p_i$
4	$p_t > 1/100 \times \sum_{i=t-1}^{t-100} p_i$
5	$p_t > 1/2[Minp_i + Maxp_i]_{i\in[t-1,t-10]}$
6	$p_t > 1/2[Minp_i + Maxp_i]_{i\in[t-1,t-100]}$

Fig. 5. Technical analysis descriptors used by our LCS agent

One of the main data contained in the order book is the gap between the best bid and the best ask, called bid-asked spread (Fig. 2). This is a common measure of the liquidity of a market, because the wider the spread is, the more important the consequences are in case of mistake in reasoning (intuitively, the cost implied by selling an asset and buying it again is higher).

In order to efficiently take advantage of the information in the order book, we propose to give our LCS agents two new kinds of descriptors. The first ones take into account the value and the evolution of the bid-ask spread, and the second ones take into account the imbalance between bid and ask. The descriptors we introduce here are evaluated in the results section.

The Bid-Ask Spread. The most intuitive approach is probably to reason on the value of the spread (descriptor 7, Fig. 6). But it does not take the scale of the values into account, that is why the proportion is a more relevant parameter. We prefer to use the ratio $r = \frac{bestP_{Ask}}{bestP_{Bid}}$ (descriptor 8).

One can, for example, compare the current value of this ratio to a previous value (descriptor 9) to determine whether the current value of the bid-ask spread is rather high or rather low. We can also use the average of the k_{10} previous values of r, or the midrange of the k_{11} previous values.

7	$bestP_{Ask} - bestP_{Bid} < k_7$
8	$r_t < k_8$
9	$r_t > r_{t-k_9}$
10	$r_t > 1/k_{10} \times \sum_{i=t-1}^{t-k_{10}} r_i$
11	$r_t > 1/2[Minr_i + Maxr_i]_{i\in[t-1,t-k_{11}]}$

Fig. 6. Descriptors based on the variation of the bid-ask Spread and the $r = \frac{bestP_{Ask}}{bestP_{Bid}}$ ratio

Imbalance between Bid and Ask. The relative size of the two parts of the order book (in quantity of assets to trade) is useful information, because it can reveal an imbalance between bid and ask (Fig. 7). Either way, this imbalance may indicate a future variation of the price, and the agent can take advantage of it. Descriptors 12 and 13 check if there are orders in each part of the order book, and descriptor 14 reasons on the ratio $\frac{nbAsks}{nbBids}$.

The relative size of each part of the order book is not the only way to evaluate imbalances between bid and ask. Indeed, if 10 assets are sold at the same price p_0, the ask is better than if 1 asset is sold at p_0 and the 9 other have a higher price, but the ratio $\frac{nbAsks}{nbBids}$ does not change. To take the price into account, we propose this measure of bid and ask :

$$ask = \sum_{order \in AskOrderBook} \frac{Quantity(order)}{Price(order) - bidAskMid}$$

$$bid = \sum_{order \in BidOrderBook} \frac{Quantity(order)}{bidAskMid - Price(order)}$$

with:

$$bidAskMid = \frac{bestP_{ask} + bestP_{bid}}{2}$$

$bidAskMid$ is the average of the best selling price and the best buying price. By dividing the quantity of each order by the difference between its price and $bidAskMid$, we take in account the fact that some orders have a limit price too low or too high to constitute an attractive bid or ask. We are interested in the ratio $q = \frac{ask}{bid}$ (descriptor 15) and its variations (descriptors 16 to 18).

12	No bid in the order book
13	No ask in the order book
14	$\frac{nbVentes}{nbAchats} > k_{14}$
15	$q_t > k_{15}$
16	$q_t > q_{t-k_{16}}$
17	$q_t > 1/k_{17} \times \sum_{i=t-1}^{t-k_{17}} q_i$
18	$q_t > 1/2[Minq_i + Maxq_i]_{i \in [t-1, t-k_{18}]}$

Fig. 7. Descriptors based on the size of each part of the order book and on the $q = \frac{ask}{bid}$ ratio

The descriptors on Fig. 6 and 7 are only examples of order-book-based descriptors, but it is of course possible to design and test others. Descriptors that use a constant k were implemented several times, with various values of this constant.

3 Methodology

Evaluating agent behaviours is a difficult task. Firstly because like for voting systems it is always possible to favour a particular agent. Secondly because an agent is rarely performant in itself, but relative to its competitors and to its environment.

In order to compare agents, it is necessary to define how the performance of an agent is measured. Two classics are possible in our case: the cash owned by an agent, and the amount of this cash with the estimated value of its portfolio. This sum is called wealth.

$$wealth = cash + \sum_{i=1}^{i<assets} price_i \times nbAssets_i$$

Even if this measure can be discussed since it is an approximation, it allows to take into account all the possessions of the agent. That is why we use this measure for our work.

Now that we have chosen a measure, there are many ways to evaluate agent performance in a group. Two main kinds of evaluation coexist [12]: evaluation in which the n agents evolve in the same environment and are competing with one another, and evaluations in which the n agents are ranked relatively to the same set of competitors.

The ecological competition is a selection method inspired by biology and natural selection ([13], [14]). Several families coevolve like animal or vegetal species sharing an environment. Like in nature, their populations vary according to time, so that the population of the best families increases, and the population of the worst families decreases.

In our case, a competition is a series of simulations where the total number of agents is constant. Each family starts with an identical number of agents. After each simulation, called "generation", the population of each family is evaluated according to its profits. In order to keep a constant total population, we have to apply a proportionality rule on the score of each family.

The score of a family is the total profits of its members during the simulation. But the individual profits of an agent can be negative, therefore the total profits of a family can be negative too, and applying a proportionality rule requires having positive values. To solve this problem, we propose to subtract from the profits of each agent a the profits of the worst agent in the simulation w.

$$profits_a \geq profits_w \Rightarrow profits_a - profits_w \geq 0$$

The modified profits of each agent is thus positive, as well as the total profits of each family. It is then possible to apply a proportionality rule to compute the new population of each family.

The total profits of family f is the sum of the modified wealth of all the agents of this family.

$$totProfits(f) = \sum_{a \in f} (profits_a - profits_w)$$

The competition has a constant total population. At the end of each generation, the population of a family is proportional to its total profits during the previous simulation.

$$pop(f) = \frac{totProfits(f)}{\sum\limits_{i \in families} totProfits(i)} totalPop$$

The population of an agent family at the end of a competition represents how well this kind of agent fits in a particular environment (the agent families in competition), and its effectiveness in this environment.

It is difficult to maintain that an agent is better than another in itself. However, if an agent obtains better results than another in several ecological competitions that are different enough to provide various environments, one can postulate that this agent is globally better. This is this method that we used to evaluate our agents.

4 Results

Many simulations have been run to obtain the results shown in this section. To conduct them, we used a specific protocol.

4.1 Experimental Protocol

In order to highlight the contribution of an order-book-based learning, in this paper we have decided to compare various kinds of LCS agents. One of these only uses a price-based LCS with the descriptors of figure 5). We use these agents as basic agents for our comparisons. Our goal is to overtake these.

We compare this basic agent type to various kinds of prices and order-based LCS agents, that use the same price-based descriptors as the basic agent, but also one or several order-based descriptors, from figures 6 and 7).

In order to generate varied situations, the competitions are populated with various agent families:

- chartist agents (moving average, RSI, momentum, variation, indicators and mixed moving average): these agents use simple conditions on the prices to forecast their future variation and decide to buy or sell
- periodic agents : periodically buy and sell
- Zero Intelligence Traders (ZIT, [15]) : these agents send orders in a random direction with a random limit price

All the learning-agent families are used in several ecological competitions, that are the same for every tested family. For each competition, the final populations of these families are compared to determine the best kind of agent for a particular environment. For each family, this operation is repeated in about ten competitions, five times per competition. Fifty generations per competition are enough to allow the population to stabilise in most cases. Each generation is equivalent to one day in which 2000 decisions are taken by each agent, enough to allow the learning agents to adapt.

4.2 Results Obtained

Figure 8 shows the average population of several agent families for various competitions. In each family, the agents have the same descriptors as the basic price-based agent, and we add to them a set of 1 to 5 order-based descriptors. We observe that several families obtain better results than the basic agent. The descriptor sets of the families presented in figure 8 are those that give best improvements. These use up to 3 new descriptors, that are instances of descriptors 10, 11, 16 and 17 (see Fig. 9). The agents that use these interesting descriptors (the other descriptors give little or no improvement) have a final population 100% greater than the basic agent. Therefore we can consider that LCS agents are really improved by learning with orders.

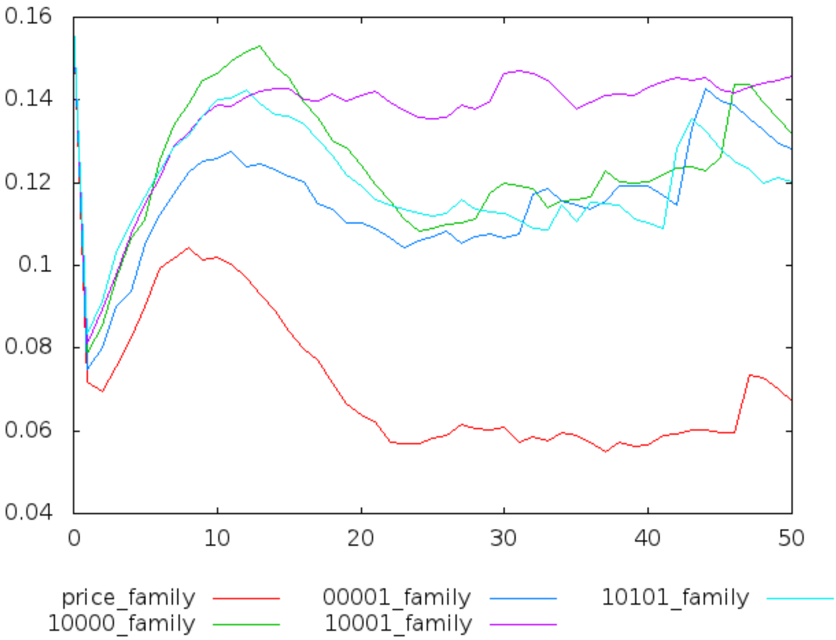


Fig. 8. Average part of the total population during generations, for the basic agent type (price_family) and various families using order-based descriptors. The binary sequence in the name of each family describes the additional descriptors used. Each bit refers to a descriptor in Fig. 9, in the same order.

	k	Descriptor's instance
10	100	$r_t > 1/100 \times \sum_{i=t-1}^{t-100} r_i$
11	100	$r_t > 1/2[Minr_i + Maxr_i]_{i\in[t-1,t-100]}$
16	5	$q_t > q_{t-5}$
17	10	$q_t > 1/10 \times \sum_{i=t-1}^{t-10} q_i$
17	100	$q_t > 1/100 \times \sum_{i=t-1}^{t-100} q_i$

Fig. 9. Order-based descriptors improving agents results

4.3 Adding Descriptors to an LCS Agent

The more information has an agent, the more likely it is to give an accurate forecast of the price variation. Nevertheless, adding a descriptor to an agent broadens its research scope, and makes the learning slower or even less efficient.

All the information does not have the same relevance. For example, we can suppose that descriptor $p_t > p_{t-1}$ is more relevant than descriptor $p_{t-100} > p_{t-101}$ in many cases. But the broadening of the search scope being the same whatever the added descriptor is, the information added by a descriptor has to be significant to offset this broadening. Moreover, if two descriptors provide similar information, the value added by the second one is very low. An LCS has to use a limited descriptor set (from 6 to 9), and these have to bring significant and varied information.

5 Conclusion

To carry out realistic financial simulations, it is important to populate artificial markets with adaptive behaviour agents.

Until now, in the absence of software platforms based on orders that use a multi-agent approach, this kind of simulation was conducted with a basic equational price-fixing model. For example this is the case in the well-known SF-ASM. The ATOM platform, with its completeness and its high relevance to order-driven markets like Euronext-NYSE, allows to improve the learning ability of agents.

In this paper, after having detailed the various possibilities to reason on the orders and their consequences, we have shown how to develop classifier systems that take into account all the system information for each agent. In order to compare these agents, we have implemented an original adaptation of an ecological competition that allows us not only to measure the performance of an agent, but also its robustness to environment modifications. Thus, we have highlighted that an agent that studies pending orders is far more efficient than its counterpart which only reasons on the prices.

Further work has to be done in this area : varying the learning methods, the descriptors, or trying to recognise the individual behaviour of the agents. We consider that this paper is a one step in the development of efficient order-based learning agents.

References

1. Arthur, W.B., Holland, J., LeBaron, B., Palmer, R., Tayler, P.: Asset pricing under endogenous expectations in an artificial stock market. In: The Economy as an Evolving Complex System II, pp. 15–44. Addison-Wesley, Reading (1997)
2. LeBaron, B.: Building the santa fe artificial stock market. Brandeis University internal report (2002)
3. Brandouy, O., Mathieu, P.: Efficient monitoring of financial orders with agent-based technologies. In: Demazeau, Y., Pechoucek, M., Corchado, J.M., Pérez, J.B. (eds.) Adv. on Prac. Appl. of Agents and Mult. Sys. AISC, vol. 88, pp. 277–286. Springer, Heidelberg (2011)
4. Mathieu, P., Brandouy, O.: A generic architecture for realistic simulations of complex financial dynamics. In: Demazeau, Y., Dignum, F., Corchado, J.M., Pérez, J.B. (eds.) Advances in PAAMS. AISC, vol. 70, pp. 185–197. Springer, Heidelberg (2010)
5. Belter, K.: Supply and information content of order book depth: The case of displayed and hidden depth (2007)
6. Cao, C., Hansch, O., Wang, X.: The informational content of an open limit order book. In: EFA 2004 Maastricht Meetings Paper (2003)

7. Kozhan, R., Salmon, M.: The information content of a limit order book: The case of an fx market. Journal of Financial Markets 15(1), 1–29 (2012)
8. Cornuéjols, A., Miclet, L., Kodratoff, Y.: Apprentissage Artificiel: Concepts et algorithmes. Eyrolles (2002)
9. Barbosa, R.P., Belo, O.: An agent task force for stock trading. In: Demazeau, Y., Pechoucek, M., Corchado, J.M., Pérez, J.B. (eds.) Adv. on Prac. Appl. of Agents and Mult. Sys. AISC, vol. 88, pp. 287–297. Springer, Heidelberg (2011)
10. Cao, L., Tay, F.: Application of support vector machines in financial time series forecasting. Omega: The International Journal of Management Science (29), 309–317 (2001)
11. Booker, L., Golberg, D., Holland, J.: Classifier systems and genetic algorithms. Artificial Intelligence 40(1-3), 235–282 (1989)
12. Beaufils, B., Mathieu, P.: Cheating is not playing: Methodological issues of computational game theory. In: ECAI 2006, pp. 185–189. IOS Press (2006)
13. Lotka, A.: Elements of Physical Biology. Williams and Wilkins Company (1925)
14. Volterra, V.: Variations and fluctuations of the number of individuals in animal species living together. ICES Journal of Marine Science 3(1), 3–51 (1928)
15. Gode, D.K., Sunder, S.: Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. Journal of Political Economy 101, 119–137 (1993)

From Real Purchase to Realistic Populations of Simulated Customers

Philippe Mathieu and Sébastien Picault

Université Lille 1, Computer Science Dept. LIFL (UMR CNRS 8022),
Cité Scientifique 59655 Villeneuve d'Ascq Cedex, France
{firstname.surname}@univ-lille1.fr
http://www.lifl.fr/SMAC/

Abstract. The use of multiagent-based simulations in marketing is quite recent, but is growing quickly as the result of the ability of such modeling methods to provide not only forecasts, but also a deep understanding of complex interactions that account for purchase decisions. However, the confidence in simulation predictions and explanations is also tightly dependent on the ability of the model to integrate statistical knowledge and the situatedness of a retail store. In this paper, we propose a method for automatically retrieving prototypes of consumer behaviors from statistical measures based on real data (receipts). After preliminary experiments to validate this data mining process, we show how to populate a multiagent simulation with realistic agents, by initializing some of their goals with those prototypes. Endowed with the same overall behavior, validated in earlier experiments, those agents are put into a spatially realistic store. During the simulation, their actual actions reflect the diversity of real customers, and finally their purchase reproduce the original clusters. Besides, we explain how such statistically realistic simulation may be used to support decision in retail, and be extended to other application domains.

Keywords: Agent-based Simulations, Knowledge discovery, Marketing, Interactions.

1 Introduction

Since several years, individual-based models and multiagent-based simulations have been used to enhance the understanding of complex marketing issues and support decision in the context of retail stores [1,2]. Due to the introduction of fine-grained information regarding individual behaviors, agent-based models are able to provide not only global predictions (such as the quantities of transactions or revenue) but also insights concerning the reasons that make marketing techniques efficient or not.

Classical marketing analysis techniques, used e.g. to segment customers into subgroups with similar needs, or to detect items that are frequently bought together, consist in retrieving global information from large databases through data mining algorithms [3,4]. For instance, association rules capture co-occurrences between items in customer baskets, and thus allow the retailer to offer promotions on frequently associated items, or to propose relevant similar products. But, since those techniques only capture statistical features of customer behavior, without being able to provide any kind of causal

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 216–227, 2013.

explanation, their quality is highly dependent on the application that uses the retrieved knowledge [5]. Besides, the data collected in retail stores result from a complex decision process, which is affected by seasonal, geographical, cultural, environmental factors, by demographic and psychographic variations of the customers, by the brand management, and by in-store events such as promotions. Thus it is difficult to assess the stability of the rules that are built from the data, and quite impossible to predict how changes in the store management affect the rules.

On the contrary, agent-based models allow to take into account individual preferences and even psychological expertise [6], so as to build an accurate description of the motivations and needs of each customer. Then it is the actions of those simulated customers that are responsible for the purchases that are predicted. Hypotheses regarding the factors that influence sales are made explicit in the model: they can be understood and examined by experts, and validated (or not) through an appropriate experimental setup.

As a counterpart, individual-based models, especially when they involve cognitive agents (e.g. for accounting psychological motivations [6]), tend to require too much expertise, which is not always easy to acquire or implement. In addition, few store simulation models do take into account the spatial issues which are considered crucial in retail stores, such as shelves allocation, items placement, checkout sizing, point of sale display, etc. In a previous work, we addressed those questions in order to build a simulation of supermarkets, where the agents were situated in a realistic environment [7]. *This situatedness is necessary for raising multiagent systems from casual, ad hoc simulations, to full decision support systems*, able to predict how the clients react to changes in the spatial organization of the store, marketing events (e.g. discount, publicity...), and competitor shops.

In this paper, we propose a simulation approach which does not rely much on expert knowledge; instead, it tries to retrieve as much information as possible from retail data (e.g. receipts) as in classical basket analysis methods; but, this knowledge discovery process is used to initialize the purchase preferences of the population of simulated customers with statistically realistic traits. Combined with a model of customer behavior, those traits produce statistically realistic purchase when the agents are acting in their environment.

Since we designed and validated a model of customer behavior in a previous work [7], we will focus in this paper on the realism of customer populations, i.e. on a purchase decision model which can mimic actual behaviors.

The main classical technique for information retrieval in actual data is the affinity analysis [3,8], which is based on the census of item co-occurrences in the purchases. It can be directly applied to actual receipts. On this basis, association rules between items (i.e. $X \rightarrow Y$ where X,Y are disjoint sets of items) can be inferred [3], together with a *support* (proportion of purchases which include both X and Y) and a *confidence* (conditional probability of buying the articles of Y when those of X are in the basket).

This approach is very helpful for cross-selling or up-selling, and to some extent it can provide indications in product placement (e.g. try to associate in the shelves items that are frequently bought together). Its first limitation is the computational time, which grows at least with the cube of the number of items [9]. But also, it is quite difficult to use association rules to drive the purchase decisions of agents, since a rule can only suggest items that are *related to others*, but not how to bootstrap the basket.

An alternative approach consists in trying to predict *shopping lists* from the existing receipts. For instance in [10] this method is implemented in a personal assistant that learns individual purchase habits, so as to remind the customers of their most probable needs during the shopping trip, and to propose personalized promotions. The purpose of this application is very far from ours, and the classification methods that are used do not build any symbolic description of the shopping list, but only a prediction over rough product categories. Nevertheless, this work assesses the possibility to perform some kind of induction over real receipts so as to identify an underlying shopping list.

In our proposal, we try to join the identification of similar customers (like in classical market segmentation) and the induction of abstract descriptors for those clusters, specifically prototypical shopping lists. Then we use the association of clusters and shopping lists to generate profiles of agents which buy close items. Hence the approach we propose follows a kind of methological loop. We define a representation frame for transactions (e.g. receipts) and their abstract description: prototypes built by generalization. Afterwards those prototypes are used to initialize the agents in the simulation and the simulated transactions can be in turn analysed.

The paper is organized as follows: section 2 presents briefly the context of the grocery store simulation that has been designed and tested in [7,11]; especially we explain how shopping lists are used to induce purchase preferences. Section 3 describes the way we represent relevant information to identify and characterize items, transactions and prototypes. Section 4 presents the data mining process that builds prototypes from transactions, and section 5 how the overall procedure has been validated. Finally we explain in section 6 how to use our approach within multiagent simulations.

2 Context of the Simulations

The knowledge discovery process we propose in this paper has been tested within an existing grocery store simulation [7,11], endowed with a realistic environment and populated with behaviorally convincing artificial customers.

2.1 An Interaction-Oriented Model

In this work, we had to acquire expertise and build the simulation model in a quite incremental and empirical way. Thus the modeling method had to be highly understandable by experts outside the field of computer science, e.g. psychologists or marketing advisors, and enable a step-by-step design of behaviors. Therefore, we used the principles of the 'Interaction-Oriented' approach [12]. In this method, each relevant entity of the model is represented by an agent, without any prior distinction between “true” agents and resources or objects; each behavior is modeled by a separated piece of code called an “interaction”, i.e. a sequence of actions between a source agent and one or more target agents, controled by some conditions. The interaction is realizable if the source and target agents fulfil the conditions.

Agents and interactions can be developed in independent libraries, then the simulation is designed by assigning interactions to pairs of agents (in an 'interaction matrix' –

see table 1). It is a visual way to express what families of agents are allowed to interact, and with what interactions. This interaction matrix is processed by a generic simulation engine, which essentially evaluates what interactions can be realized, between what agents, and subsequently determines the actions to be performed by each agent.

Actually, describing the action capabilities of the entities of the model in terms of interactions that can be performed or undergone, instead of behaviors embedded in the agents, is very close to the theory of affordances [13]. Thus, it makes this quite natural to use for psychologists and facilitates knowledge acquisition.

2.2 Model of a Customer Agent

In the work presented here, we use a model of customer behavior that was previously developed for the simulation of a retail store, aimed at studying human vendors confronted to artificial clients [11]. Thus, the overall behaviors of simulated customers have been validated by marketing experts and are set once and for all in the work described here. In what follows, the vendor was removed for studying only the clients. The corresponding interaction matrix is shown on table 1.

Table 1. The interaction matrix that defines the behavior of all agents in our simulation. For instance, the intersection between line 'Customer' and column 'Item' contains two interactions, 'Take' and 'MoveTowards', which means that a customer agent may either take an item agent, or move close to it, depending on the priority (first number – here, taking an item has the highest) and on the distance between agents (second number), assuming that the conditions of the interactions are fulfiled for both agents. The ∅ column contains reflexive interactions (i.e. where the target agent is the source itself). Empty columns and lines have been removed.

Source \ Target	∅	Customer	Item	Checkout	Queue	Door
Customer	Wander (0) GoToPlace (1, ∞)		MoveTowards (2, 10) Take (4, 1)	MoveTowards (3, 10)	StepIn (5, 2) MoveOn (6, 1) WalkOut (7, 1)	Exit (8, 1)
Item		Notify (1, 10)				
Sign		Notify (1, 10)				
Checkout	Open (10) Close (10)	Notify (1, 15) CheckOut(7, 1)			Handle (8, 1) ShutDown (9, 1)	
Door	SpawnCustomer(1)	Notify(1, 10)				

To summarize, it is assumed here that all clients have the same overall behavior, but *differ in their needs*: thus they are endowed at startup with a *shopping list*, which specifies more or less precisely what items they are likely to buy in the store. The needs may be specified with accuracy, e.g. "SodaCola light, family pack", or with vague indications, e.g. "spring water", which can match much more actual items. The purchase decision is implemented by the 'Take' interaction, which consists of two conditions (the target agent matches an item of the shopping list of the source agent; and: the source has enough money) and one action (put the target in the basket of the source).

During the simulation, interactions occur according to the interaction matrix and the state, perceptions and positions of agents, producing a consistent consumer behavior: artificial clients try to find all items which figure in their list, within a limited amount of time. They may be endowed with a mental map of the shop or with no prior knowledge;

they also are notified by panels, checkouts, items, etc. about relevant information to help them.

In the original work, the shopping lists were either computed through a pure random process (following a Poisson distribution based on the average basket size), or implemented "by hand", according to a specific mise en scène, in a deliberate intend to put the vendor in a problematic case. In the experiments described below (see section 5), those lists have been built through the knowledge retrieval algorithm we propose.

3 Knowledge Representation

Before describing the data mining process which builds prototypes from transactions, we must explain how we represent both kinds of information. This step may require the intervention of a marketing expert, but afterwards the knowledge retrieval process is automatic.

3.1 Items Identifiers: From SKUs to Meaningful Descriptions

In retail stores, each unique product is usually identified by a "Stock-Keeping Unit" (SKU) in order to track availability and demands. The SKU does not necessarily carry any special meaning regarding the nature and characteristics of the product. Other methods, such as the Universal Product Code (UPC), European Article Number (EAN), etc. can be used as well. It may also happen that the actual purchase are anonymised through an automatically generated identifier, e.g. in order to perform basket analysis under strong confidentiality constraints.

Those identification methods are actually not well suited for extracting more than co-occurence rules. Relevant marketing knowledge (e.g. product family, quality, relative price, brand image, organic label...) must be added to characterize the products so as to allow an explanatory analysis.

In our approach, each unique product is identified by a **tuple of strictly positive integers**, which encodes the features values that are considered relevant in the application context. For instance, if the relevant features are the brand, the product family and the details (e.g. respectively "SodaCola", "beverage", "soda with cola"), then products will be identified only by a triple of integers, e.g. `(31, 4, 15)`. This allows a representation of all products at an arbitrary fine level, including specific labels such as "organic", "fair trade" or "gluten-free". Also, continuous values such as the price or weight may be encoded through a prior categorization (e.g. 1 for "cheap", 2 for "average", 3 for "expensive"; or 1 to 4 for small to extra large packings). To some extent, the mapping between SKU (or other identification systems) to this kind of integer tuple can be performed automatically, through an appropriate join of databases. Yet, the selection of features that have to be taken into account for providing a relevant description of the products may involve a marketing expertise.

3.2 Transactions and Prototypes

Our data mining process relies upon the recording of actual purchases. The simplest way to do this is to retrieve information directly from the receipts. A *transaction* can be computed as a mere enumeration of the unique products of a receipt. Quantities are not taken into account as such (exactly like in the classical affinity analysis process), though they could be added as a trait of each item (like other continuous features, see above). Thus, from a list of SKU, we build a set of integer tuples.

From those transactions, the knowledge retrieval process consists in building *prototypes*, which are aimed at describing an "abstract receipt" so as to characterize clusters of receipts. In order to do so, we introduce the concept of *prototype item*, which are also integer tuples, but **allowing the value 0 as a wildcard**. For instance, a product characterized as "any brand", "beverage", "soda with cola", could be described by the triple `(0, 4, 15)`. The null tuple `(0, 0, 0)` means "any item". A *prototype* is simply a set of prototype items.

Those prototypes, built from real data, may also be used as a "shopping list" for simulated customers, because it may often happen that only few traits of the desired items are specified. For instance, Mr Smith always buys "soda with cola" but is indifferent to the brand, while Mr Wesson is likely to buy any organic yoghurts from the brand "Yoopla". The use of the 0 wildcard is very helpful for expressing such vague wishes.

In the next section, we show how such prototypes are actually built from the transactions.

4 Steps of the Data Mining Process

In order to analyse the purchases, we proceed as follows:

1. The transactions database is partitioned into clusters (this requires first to define a distance measure between receipts, which is itself based on a distance measure between items).
2. For each cluster:
   (a) all items that appear in the transactions are in turn classified so as to build prototype items ;
   (b) the prototype composed of the union of prototype items is scored against the transactions of the cluster.

4.1 A Measure of Item Similarity

Since some items of a customer's shopping list may be not fully specified, it is expected that several customers who have the same prototype (i.e. the same shopping list) will not get exactly the same items. Thus, if the distance between transactions relies only upon the equality of items, it is likely to produce a defective clustering. Instead, we propose to modulate the comparison between transactions, by taking into account the distance between items.

A simple way to do so is to compute a Hamming-like distance (or conversely, a Hamming-like similarity index). If two items are encoded by the tuples $I = (f_1, \dots, f_n)$ and $I' = (f'_1, \dots, f'_n)$, the similarity between items is defined as: $\sigma(I, I') = \frac{1}{n}\sum_{i=1}^{n} \delta(f_i, f'_i)$ where $\delta(f_i, f'_i) = 1$ if $f_i = f'_i$ or $f_i = 0$ or $f'_i = 0$, and 0 otherwise (note that this definition works fine for actual items and for prototype items as well).

4.2 A Measure of Transaction Similarity

In order to compare transactions, we started with a well-known measure which is frequently used for measuring similarities between sets: the Jaccard index [14]. It is defined for any subset X,Y as follows: $J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} = \frac{|X \cap Y|}{|X|+|Y|-|X \cap Y|}$

As it has been previously said, the rough use of the Jaccard index cannot fit our needs, since it would make no difference between disjoint sets and sets that contain very similar but different items. Thus we propose an extension of the Jaccard index, based on the distance between items. It consists in computing the best matching score between items of both transactions (thus we called it the *best-match Jaccard index*, denoted by J_{BM}). To compute it between a transaction $T = \{I_1,...,I_p\}$ and another transaction $T' = \{I'_1,...,I'_q\}$, we follow these steps:

1. Compute the matching matrix $(\sigma_{i,j})$ with $\sigma_{i,j} = \sigma(I_i, I'_j)$
2. For each k between 1 and $\min(p,q)$:
   (a) compute $\mu_k = \max_{i,j}(\sigma_{i,j}) = \sigma_{i^\star,j^\star}$ (if several (i,j) values verify $\mu_k = \sigma_{i,j}$, we take one pair $(i^\star, j^\star)$ which minimizes: $\left(\sum_{i \neq i^\star} \sigma_{i,j^\star} + \sum_{j \neq j^\star} \sigma_{i^\star,j}\right)$)
   (b) replace $(\sigma_{i,j})$ by the submatrix obtained by deleting row $i^\star$ and column $j^\star$
3. $\mu_{BM} = \sum_{k=1}^{\min(p,q)} \mu_k$ plays the same role as $|X \cap Y|$ in the classical Jaccard index, so we have: $J_{BM}(T,T') = \frac{\mu_{BM}}{p+q-\mu_{BM}}$

For example, we take $T =\{$`(1, 1, 2), (3, 5, 8), (13, 21, 34)`$\}$ and $T'=\{$`(1, 1, 2), (3, 6, 8), (12, 13, 14), (1, 1, 34)`$\}$. Since $T \cap T' =\{$`(1, 1, 2)`$\}$ only, we have $J(T,T') \approx 0.1666667$, while $J_{BM}(T,T') = 0.4$ because several items of T and T' are close.

A large number of similarity and distance measures can be used as well [15,16]; actually, the method we propose is also suitable for frequently used measures, such as Ochiai [17] or Sørensen-Dice [18], which can be extended using the same best matching algorithm as we did for Jaccard. This point was checked experimentally through the same procedure as we present in section 5.

4.3 Transactions Clustering

We apply the best-match Jaccard index for computing a distance matrix between all transactions in the database: $\Delta_{BM} = (\mathrm{d}_{i,j})$ with $\mathrm{d}_{i,j} = 1 - J_{BM}(T_i, T_j)$. The distance matrix can be used with a large number of clustering techniques; we chose a very classical hierarchical clustering algorithm, namely that implemented in the `flashClust` library in R [19], which easily provides dendrograms w.r.t. the similarity between transactions.

Then, the dendrogram can be cut into K classes (e.g. with the R function `cutree` [20]). The appropriate value of K is far from obvious, so we tried to evaluate empirically the appropriate height to cut the tree from randomly generated prototypes, where the value of K was known (see section 5). We found that a height $h \approx 0.6$ was quite convenient in all cases.

4.4 Prototype Induction

Building a prototype for a class, means essentially finding a set of integer tuples (with zeros allowed) which has the highest matching value with the N transactions of the class w.r.t. the best-match Jaccard index. We start with a frequency analysis, i.e. for each item I that appears on some transactions of the given class, we compute $f(I)$ as the ratio between the number of transactions that include I and N.

Rare items, i.e. with $f(I) < \varepsilon$, may be considered casual purchase, or "noise", and simply discarded (typically this works fine with $\varepsilon \approx \frac{1}{N}$). Conversely, very frequent items, i.e. with $f(I) > \theta$, may be considered a must-have and kept unchanged in the prototype (typically $\theta \approx 0.95$).

Regarding intermediate items, we have to classify them again, in order to be able to detect that e.g. "SodaCola" products are always associated with "organic yoghurts" of several brands. Thus we compute a matrix distance between items: $(D_{i,j})$ with $D_{i,j} = 1 - \sigma(I_i, I_j)$ and use it for building a dendrogram of the items. There again, the number of classes K_I is not known a priori.

Therefore, we iterate the following process for several possible values of K_I:

1. for each cluster: build a prototype item by putting zeros where features differ ; for instance, if the items in the cluster are `(1, 5, 7)`, `(1, 6, 7)` and `(1, 12, 7)`, then the prototype item is `(1, 0, 7)`
2. collect all prototype items and join them with the very frequent items (see above) to build the candidate prototype P_{K_I}
3. compute the score of K_I as the average value of $J_{BM}(P_{K_I}, T)$ for all transactions T in the original class.

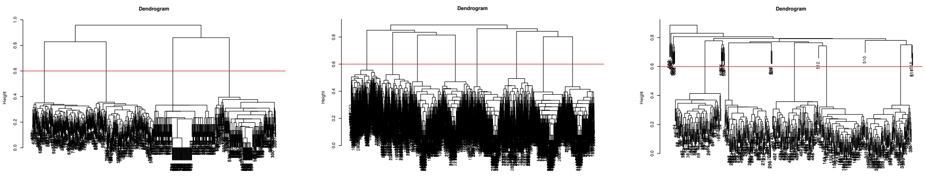
Finally, we keep the value $K_I^\star$ (and associated prototype $P_{K_I^\star}$) which maximizes this score.

5 Validation of the Data Mining Process

As explained before, prototypes are used in a simulation process to produce artificial transactions. Since the agents perform autonomous behaviors, according to their shopping lists which contain prototype items (i.e. with wildcards), and in the situated context of a realistic store with possibly missing (or hard to find) items, the transactions that occur in the simulation are not expected to be exactly the same than the real ones.

Yet, we have to assess that the simulation is able to reproduce the same kind of customer behavior that is observed in reality. Therefore we can analyse the transactions produced by the simulation, build the corresponding prototypes through the same data mining process, and compare them to the prototypes that resulted from real data.

However, a first step towards analysing the outcome of the multiagent simulation is to prove the robustness of our prototype building method. Otherwise, possible differences between the prototypes that reflect the activity of the agents and the original ones, could not be explained by the behavior of the agents or the characteristics of the environment, but maybe just by a high sensitivity of the data mining process to perturbations. Thus we present here how the robustness of our analysis method has been tested.



(a) (b) (c)

Fig. 1. Dendrograms built according to the similarity of transactions in 3 experiments. Parameters: (a) $N_P = 4, N_I = \{5, 10, 20, 40\}, N_T = 200, N_A = 5\ \%, N_M = 5\ \%, N_O = 0$; (b) $N_P = 8, N_I = 10, N_T = 400, N_A = 5\ \%, N_M = 5\ \%, N_O = 0$; (c) $N_P = 5, N_I = 20, N_T = 100, N_A = 5\ \%, N_M = 5\ \%, N_O = 5\ \%$. The horizontal line represents the cut height (0.6).

5.1 Stochastic Simulations of Prototypes Instantiation

In order to perform extensive tests, we generated several sets of prototypes, each one containing random prototype items. Then, in order to obtain a coarse-grained, but quick, simulation of the purchase induced by such prototypes, we instantiated them through a stochastic process, based on the following parameters:

- the number N_P of prototypes to test
- the number of prototype items $N_I(i)$ in each prototype i
- the number of transactions per prototype, $N_T(i)$, which determines how many transactions are instantiated for prototype i
- the number of additional random items $N_A(i)$ which indicates how many randomly chosen items are added to transactions of prototype i (we do so to represent e.g. casual or compulsory purchase which are not necessarily related to the prototypical purchase habits)
- the number of missing items $N_M(i)$ which indicates, in prototype i, how many prototype items will be kept uninstantiated (this offers the possibility that some items may not be found or not be needed)
- the number N_O of transactions that do not belong to any prototype (and generated completely randomly).

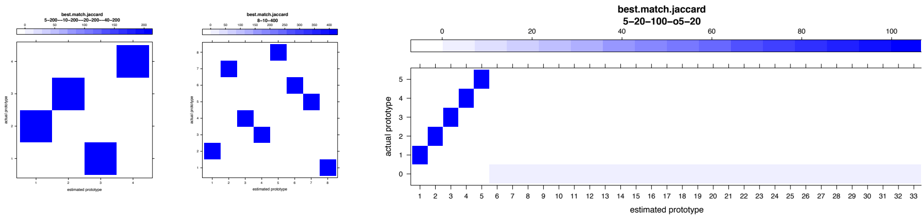
The "instantiation" of a prototype item consists in replacing each zero by a random, strictly positive, value. Thus this operation depends on the *domain* of the tuples that describe concrete items, i.e. the maximum values allowed for of each integer in the tuple. In our experiments we used 5-element tuples with maximum values `(20, 100, 10, 5, 2)` (on the basis of earlier work [11]).

We conducted automatic experiments and evaluations with a combination of all parameters within the following ranges: N_P: 4 – 10; N_I : 5, 10, 20, 40; N_T: 50, 100, 200, 400, 800; N_A: 0, 5, 10 % of items in the transactions; N_M 0, 5, 10 % of items in the transactions; N_O: 0, 5, 10 % of total transactions.

5.2 Results and Discussion

As figure 1 shows on three experiments, transactions produced by the instantiation of random prototypes are well discriminated: cutting the trees at height ≈ 0.6 is sufficient to identify clusters that exactly reflect the original ones (cf. fig. 2a-b), even when transactions are built with random additional items or with random missing items.

When the database also contains random transactions, i.e. which do not come from any existing prototype, the clustering still identifies the original clusters, but also very small classes (see fig. 2c), which are most of the time singletons. When applying the induction process, those classes can be simply discarded because there is nothing to generalize in them. In all experiments (up to $N_O = 10$ % of the total number of transactions) the prototype building process was successful, which indicates enough robustness.



(a) (b) (c)

Fig. 2. Level plot comparison between estimated transactions clusters (abscissae) and original prototypes (ordinates) for experiments (a), (b) and (c). While (a) and (b) match perfectly, in (c) there are "noise" transactions (prototype "0") generated from pure random choice. It appears that they are *not* classified among the "true" clusters, but instead are put in small classes (here, 1 for each random transaction).

6 Experimental Setup for Multiagent Simulations

At the present time, the multiagent simulator designed in [11] has been modified so as to represent shopping lists with prototypes and the items by integer tuples.

We have conducted preliminary experiments with the same randomly generated prototypes than in stochastic simulations. For now, the simulated transactions reproduce the same prototypes.

However, this result is dependent on the *time limit* which is given to the agents for their shopping trip. If too short, they exit the store without purchasing all items of their list. This does not affect the transaction clustering (because of its robustness towards missing items) but may alter the prototypes that are built from the simulated transactions. Indeed, the observation of the paths of the customers in the store points several "hot areas" where items are easily found: conversely, missing items are often the same (contrary to what happened in stochastic simulations), thus the corresponding prototype is a subset of the original one.

Far from being a limitation of the system, this property gives insights on how such kinds of simulation may help the placement of products, the spatial organization of the store, etc.: the limitation of time spent in the shopping trip is crucial, not only for the purpose of realism, but more significantly because it appears as the criterion that forces the store manager to optimize the positioning of products.

Ongoing work focuses now on the analysis and integration of large databases of real receipts. We are also modifying the environment in order to reproduce the store where the data were collected. We have for instance to integrate the actual positioning of items and information signs in the store. Indeed, the correctness of those informations may have a serious impact on the outcomes of the simulations, so we have to check them carefully and validate them with experts before we can start large-scale simulations.

7 Conclusion

The design of an integrated tool for decision support in the field of grocery retail and marketing is a long-term purpose indeed. However, in this paper we try to combine an incremental simulation approach (which is quite convenient to express complex hypotheses regarding individual behaviors, environmental configuration, etc.) with data mining algorithms (which usually indicates global, statistical features of a system). Our proposal is therefore able to endow agents populations with statistically realistic features, which in turn affect the behavior of the agents so as to produce statistically similar outputs. Far from being only qualitative, we show that this similarity can be measured. As shown above, our process is quite robust to noise in data. The results of the integration of real receipts in the multiagent simulation (still in progress) will be described in further publications.

Noteworthy, our method does not try to discover "true" classes of customers, such as a socio-economic, or demographic, or geographic segmentation would aim at. We only intend to capture similarities between traces left by individual actions, and use an abstract description of those traces as parameters of agents behaviors. Thus, taking into account geographic influences or seasonal variations merely relies upon an appropriate choice of the recording extent and duration for the real data.

Besides, we believe (though we cannot provide experimental evidences yet) that the transaction and prototype representation we propose can be applied to many other fields where the behavior of the entities can be characterized by such sets of features (e.g. molecular biology with phenotypical expressions of co-activated genes, ecology, or auction management), so as to participate in the bootstrap of multiagent simulations from empirical data.

References

1. Schwaiger, A., Stahmer, B.: Simmarket: Multiagent-based customer simulation and decision support for category management. In: Schillo, M., Klusch, M., Müller, J., Tianfield, H. (eds.) MATES 2003. LNCS (LNAI), vol. 2831, pp. 74–84. Springer, Heidelberg (2003)
2. Siebers, P.O., Aickelin, U., Celia, H., Clegg, C.W.: Using intelligent agents to understand management practices and retail productivity. In: Proceedings of the Winter Simulation Conference (WSC 2007), Washington, D.C., pp. 2212–2220 (December 2007)
3. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the 13th ACM SIGMOD Conference, Washington, D.C. (May 1993)
4. Mladenić, D., Eddy, W.F., Ziolko, S.: Exploratory analysis of retail sales of billions of items. In: Goodman, A., Smyth, P. (eds.) Frontiers in Data Mining and Bioinformatics. The 33rd Symposium on the Interface of Computing Science and Statistics (2001)

5. Sheth-Voss, P., Carreras, I.E.: How informative is your segmentation? a simple new metric yields surprising results. Marketing Research, 9–13 (Winter 2010)
6. Zhang, T., Zhang, D.: Agent-based simulation of consumer purchase decision-making and the decoy effect. Journal of Business Research 60(8), 912–922 (2007); Complexities in Markets Special Issue
7. Yoann, K., Mathieu, P., Picault, S.: An interaction-oriented model of customer behavior for the simulation of supermarkets. In: Huang, X.J., Ghorbani, A.A., Hacid, M.S., Yamaguchi, T. (eds.) Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2010), pp. 407–410. IEEE Computer Society (2010)
8. Han, J., Kamber, M.: Data mining: concepts and techniques. Elsevier (2006)
9. Cavique, L.: A scalable algorithm for the market basket analysis. Journal of Retailing and Consumer Services 14(6) (November 2007)
10. Cumby, C., Fano, A., Ghani, R., Krema, M.: Predicting customer shopping lists from point-of-sale purchase data. In: Proceedings of the 10th International Conference on Knowledge Discovery and Data Mining (KDD 2004), pp. 402–409. ACM, New York (2004)
11. Mathieu, P., Panzoli, D., Picault, S.: Virtual customers in an agent world. In: Demazeau, Y., Müller, J.P., Rodríguez, J.M.C., Pérez, J.B. (eds.) Advances on PAAMS. AISC, vol. 155, pp. 147–152. Springer, Heidelberg (2012)
12. Kubera, Y., Mathieu, P., Picault, S.: IODA: an interaction-oriented approach for multi-agent based simulations. Journal of Autonomous Agents and Multi-Agent Systems, 1–41 (2011)
13. Gibson, J.J.: The Ecological Approach to Visual Perception. Hillsdale (1979)
14. Jaccard, P.: Étude comparative de la distribution florale dans une portion des alpes et du jura. Bulletin de la Société Vaudoise des Sciences Naturelles 37, 547–579 (1901)
15. Choi, S.S., Cha, S.H., Tappert, C.C.: A survey of binary similarity and distance measures. Journal of Systemics, Cybernetics and Informatics 8(1), 43–48 (2010)
16. Boriah, S., Chandola, V., Kumar, V.: Similarity measures for categorical data: A comparative evaluation. In: Proceedings of the SIAM International Conference on Data Mining (SDM 2008), pp. 243–254. SIAM (2008)
17. Ochiai, A.: Zoogeographic studies on the soleoid fishes found in Japan and its neighbouring regions. Bulletin of the Japanese Society for Fish Science 22, 526–530 (1957)
18. Dice, L.R.: Measures of the amount of ecologic association between species. Ecology 26(3), 297–302 (1945)
19. Langfelder, P., Horvath, S.: Fast R functions for robust correlations and hierarchical clustering. Journal of Statistical Software 46(11), 1–17 (2012)
20. Becker, R.A., Chambers, J.M., Wilks, A.R.: The New S Language. Wadsworth & Brooks/Cole (1988)

REAGENT: Reverse Engineering of Multi-Agent Systems

Ricardo Pérez-Castillo[1], Juan Pavón[2], Jorge J. Gómez-Sanz[2], and Mario Piattini[1]

[1] Instituto de Tecnologías y Sistemas de Información (ITSI), University of Castilla-La Mancha
Paseo de la Universidad, 4 13071, Ciudad Real, Spain
{ricardo.pdelcastillo,mario.piattini}@uclm.es
[2] Department of Software Engineering and Artificial Intelligence,
Universidad Complutense de Madrid
Ciudad Universitaria s/n, 28040, Madrid, Spain
{jpavon,jjgomez}@fdi.ucm.es

Abstract. Agent-based technology is being used in an increasing variety of applications and domains. Despite the substantial research effort on methodologies for analyzing, designing and implementing multi-agent systems (MAS), maintenance and evolution of MAS software is nowadays challenging. This paper presents REAGENT, a reverse engineering technique for retrieving MAS design models from the source code of Jade based MAS implementations. REAGENT support tools have been experimentally evaluated by means of a case study with 19 benchmark programs from textbook. REAGENT has proved to be able to collect accurate and complete agent models in a linear time regarding the size of agent models, which facilitates its applicability to large, complex, industrial MAS.

Keywords: Multi-Agent System, Reverse Engineering, Maintenance, REAGENT.

1 Introduction

Multi-Agent Systems (MAS) are being used in an increasing variety of applications, ranging from comparatively small systems for personal assistance to open, complex, mission-critical systems for industrial applications such as process control, system diagnostics, manufacturing, transportation logistics and network management.

Similarly to traditional information systems, MAS have to be maintained in order to solve or prevent faults, and evolve adding new functionalities or meeting new requirements. During software maintenance, sometimes the only reliable information is embedded in the source code since the documentation is missing or outdated. Reverse engineering of source code aims at creating high-level representations for the existing software system to support its comprehension and evolution [5]. Program comprehension is time-consuming and entails the largest portion of the maintenance effort, especially because it is often performed manually with simple general-purpose tools, such as editors and regular expression matchers. There are also some reverse engineering tools for traditional information systems, which can facilitate the task, for example to

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 228–238, 2013.

extract facts from source code/binaries, execution traces, or historical data, to query the extracted facts, and to build high-level views of the software system (e.g., UML diagrams) [4].

While reverse engineering of information systems has been widely addressed in literature and is a mature area, there are no similar achievements on research about reverse engineering of MAS software [10]. One of the reason of this fact is the nature of MAS (e.g., communication and interaction protocols, belief sharing and propagation, behavior adaptation, etc.), which entails particular challenges making it difficult to address reverse engineering of MAS [2].

Despite the mentioned challenges, there are some approaches for reversing MAS, such as *Moreno et al.* [10], which uses natural language processing, *Bosse et al.* [3], which records agent activities by considering execution traces of MAS, or *Sauvage* [15], which employs design patterns to collect and express agent concepts. Unfortunately, most current reverse engineering techniques are *ad hoc* solutions that only focus on a particular agent framework. This entails a lack of standardization, and therefore, a lack of automation. This problem makes it difficult to reuse these techniques with large and complex MAS.

The main contribution of this paper is REAGENT, a reverse engineering technique for obtaining design models by analyzing the source code of MAS. REAGENT takes JADE-based systems as input and provides abstract design models according to INGENIAS [12], an agent-oriented development methodology. Although REAGENT currently works only with JADE and INGENIAS as the operational/execution environment, ideas from this approach can be adapted to other agent platforms and methodologies. JADE has been chosen because there is a vast number of MAS that are implemented with this platform, as it follows the FIPA standard. INGENIAS has been chosen for MAS modeling because it provides a set of tools that are based on metamodeling techniques, which facilitate the editing and transformation of models. The conformance with these well-known platforms together with the proposal is supported by a tool that ensures the reuse of this technique to large, complex and industrial MAS. This paper additionally provides experimental results about the applicability of REAGENT to a set of example JADE programs [2] that have been used as benchmarks in this context.

The remaining of the paper is organized as follows: Section 2 summarizes related work. Section 3 presents the reverse engineering approach for recovering MAS models that are specified with the INGENIAS modeling language. Section 4 shows the multi-case study with MAS benchmarks. Finally, Section 5 discusses conclusions and future work.

2 Related Work

There is some precursory work in the literature about reverse engineering of MAS. For example, *Moreno et al.* [10] have applied natural language processing techniques throughout the entire lifecycle of MAS in order to obtain a design model from existing MAS. In the same line, *Hirst* [8] has proposed analysis of pre-existing Soar agents by automatically reverse engineering them. This has been shown useful for maintaining the agents themselves, and also suggests the possibility of knowledge reuse across agent architectures. The limitation of this approach is that it proposes an *ad hoc* technique specially developed for Soar architectures.

While previous approaches mainly focus on static analysis, i.e., a syntactic analysis of source code line by line, other approaches are based on the dynamic analysis. *Bosse et al.* [3] uses the dynamic analysis of MAS by recording agent activities as execution traces to verify that the traces satisfy some specified properties. This approach ensures that the user's comprehension of the system behavior is accurate with respect to the execution traces detecting anomalous behavior. *Adra et al.* [1] have proposed a dynamic inference mechanism for the refinement of agent-based model specifications that helps to establish a confidence level about the implemented model and reveals discrepancies between observed and normal or expected behavior.

Other works focus on the migration of traditional information systems to MAS. In this line, *Chen et al.* [6] developed an evolutionary approach to reengineer legacy systems into agent-based Web services.

Studies about reverse engineering of MAS are often *ad hoc* techniques without empirical validation. As a result, the above mentioned methods are difficult to be reused with different MAS, and, therefore, their applicability to the industry cannot be ensured. To mitigate this threat, *Regli et al.* [13] made an effort to provide a reference model for agent-based systems. The purpose of a reference model is to provide a common conceptual basis for comparing systems and driving the development of software architectures and other standards. As part of this study, the team applied software reverse engineering techniques to perform static and dynamic analysis of operational agent-based systems, which enabled identification of key common concepts across different agent frameworks. Similarly, *Sauvage* [15] proposed a set of design patterns to collect and express agent concepts, which can be adapted to various MAS developing challenges. The agent patterns cover all the development stages, from analysis to implementation, including reengineering through *antipatterns*.

Despite recent software engineering paradigms like Model-Driven Development (MDD) could help with the formalization and automation of reverse engineering techniques, it is rarely incorporated to current approaches. An exception is the approach proposed by *Warwas et al.* [16] which presents a model-driven reverse engineering approach for lifting the underlying design of implemented MAS to a platform independent level. For this purpose the proposal provides conceptual mappings from the platform to a platform independent modeling language. The extracted structures can be re-used as blue print for solving similar problems on similar execution platforms. MDD has been used as well for reverse engineering associated to modernization of software systems in the MOMOCS project [7], but without considering MAS.

In summary, current approaches address testing and agent verification or focus on the analysis and measurement of MAS. So far, there are no mechanisms to abstract code-level MAS to high-level design models.

3 Reverse Engineering of JADE-based Systems

REAGENT supports reverse engineering of JADE-based MAS into a high-level design that is represented using the INGENIAS agent-oriented modeling

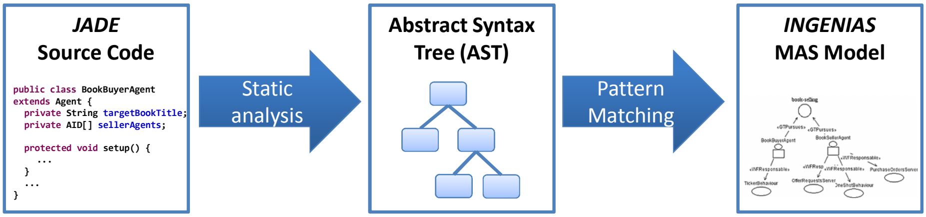


Fig. 1. The REAGENT overview: stages and artifacts involved

language. This is done in two steps, which are illustrated in Fig. 1: a static analysis of source code followed by a pattern matching recognition, which will provide a MAS model.

3.1 Static Analysis

The first stage of REAGENT employs a static analysis of JADE-based MAS implementation. Static analysis consists of the syntactic analysis of source code line by line. Static analysis can be easily carried out by means of parsers that recognize certain syntax depicted by metamodels of grammars. In comparison to dynamic analysis, which considers runtime information, static analysis is not able to take into account actual values for program variables. Nevertheless, the static analysis is more exhaustive than the dynamic analysis, since static analysis considers all parts of source code while the dynamic analysis takes into account only those parts that were reached in a particular execution. Additionally, dynamic analysis is more time-consuming than static one due to dynamic analysis firstly records traces during execution and secondly examines traces to extract relevant information.

The static analysis in this stage takes as input JADE-based source code, which is, in Java source code. REAGENT builds an abstract syntax tree (AST) using the analysis of JADE source code. AST is an intermediate representation of source code in a tree structure which can be used to easily recognize some particular structures in the pattern matching stage (see Fig. 1). The static analysis was implemented as an extension of the *Java 1.5* parser developed with JavaCC [11], a parser generator tool.

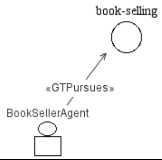
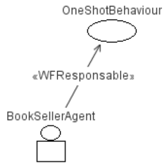
3.2 Pattern Matching

Having obtained an AST from source code, REAGENT applies a pattern matching technique for generating agent models according to the INGENIAS methodology. Pattern matching checks a perceived sequence or tree structure of tokens form the AST for the presence of the constituents of a certain pattern. Pattern matching includes the definition of the respective output structures which will be built in the outgoing agent model.

REAGENT provides three main patterns: agents, goals and tasks. Table 1 shows an example of the result of pattern matching applied to the *BookSellerAgent* class, which corresponds to one of the benchmark programs used in the case study (cf. Section 4). These patterns correspond with core concepts of the INGENIAS methodology [12].

INGENIAS provides a notation for modeling MAS and a well-defined collection of activities to guide the development of an MAS in the tasks of analysis, design, verification, and code generation, supported by an integrated set of tools. An MAS in INGENIAS is specified from different viewpoints: organization, agent, goals and tasks, interaction, and environment. REAGENT focuses on the agent model, which depicts the agents of an MAS, their goals and the tasks for which they are responsible.

Table 1. A running example for the pattern reconginition [2]

<table>
<tr><th></th><th>JADE Code</th><th>INGENIAS Agent Model</th></tr>
<tr><td>Agents</td><td><pre>public class BookSellerAgent extends Agent {
  …</pre></td><td>
</td></tr>
<tr><td>Goals</td><td><pre>public class BookSellerAgent extends Agent {
  ...
  DFAgentDescription dfd = new DFAgentDescription();
    dfd.setName(getAID());
    ServiceDescription sd = new ServiceDescription();
    sd.setType("book-selling");</pre></td><td>
</td></tr>
<tr><td>Tasks</td><td><pre>public class BookSellerAgent extends Agent {
  public void updateCatalogue(final String title,
  final int price) {
    addBehaviour(new OneShotBehaviour() {
      public void action() {
        ...
      }
    });
  }</pre></td><td>
</td></tr>
</table>

Agents attempts to identify Java classes that extend the *jade.core.Agent* class. The list of the *extends* clauses of a class in an AST is provided by the path *CompilationUnit > PackageDeclaration > TypeDeclaration > ClassOrInterfaceDeclaration > ExtendsList*. This pattern builds an Agent element in the agent model for each recognized agent class (see Table 1).

Goals searches for *DFAgentDescription* instances which are added to the yellow pages with a concrete *ServiceDescription*. The service description label is used to provide the name of the goal, which is pursued by the respective agent (see Table 1).

Tasks search for different behaviors of agent classes to figure out the tasks of each agent element in the INGENIAS agent model. This pattern searches for calls to the *addBehaviour* method of the *jade.core.Agent* class, which adds a subclass of the *jade.core.behaviours.Behaviour* class. A task is built for each recognized behavior.

4 Empirical Validation

This section presents a case study by using the example JADE programs provided in [2], which are considered as benchmark programs in this context. The study applies the proposed technique to these benchmarks to obtain design MAS models. The case study was conducted according to the formal protocol proposed by *Runeson et al.* [14] for designing, conducting and reporting case studies in the software engineering field.

The following subsections show the adaptation to MAS of the stages of this protocol: design, execution procedure, data collection, and analysis and interpretation.

4.1 Design

The *object of study* is the reverse engineering technique proposed in Section 3, while the *purpose of this study* is the assessment of specific properties of the proposed technique such as their effectiveness and efficiency. Taking into account the object and purpose of the study, two main research questions are defined. Q1 is related to the effectiveness evaluation, whilst Q2 is associated with the efficiency assessment.

Q1. Is the technique able to retrieve accurate and complete agent design model from existing MAS?
Q2. Is the technique scalable to larger and more complex MAS?

Concerning the formal design, the study is considered a multi-case study since it focuses on various agent programs. The study also follows a *holistic* design since it is applied to each case as a whole, and does not consider several analysis subunits. As a result, MAS programs are considered as the independent variable.

In order to quantitatively answer the research questions, various measures are considered as dependent variables. On the one hand, to evaluate effectiveness according to Q1, recall and precision are used. Recall is a measure of completeness whereas precision can be seen as a measure of exactness or fidelity. Recall (1) represents the number of relevant elements retrieved as a function of the total of relevant elements (retrieved and not retrieved) depicting the MAS trough an agent model. Precision (2) represents the number of relevant elements retrieved within the set of retrieved elements in an agent model. Together with recall and precision, F-measure (3) is considered for aggregating precision and recall values into a sole value by means of a harmonic mean. F-measure is necessary, since there is an inverse relationship between precision and recall. On the other hand, the study measures the time spent on the execution of each transformation so that Q2 can be answered.

Apart from these measures, other measures to characterize the input of MAS programs are used, like the number of Java files, lines of source code (LOC) and cyclomatic complexity (4). The cyclomatic complexity is the number of linearly independent paths through the source code, and is related to the intricacy of source code.

$$P = \frac{|\{relevant\ recovered\ tasks\}|}{|\{recovered\ tasks\}|} \quad (1)$$

$$R = \frac{|\{relevant\ recovered\ tasks\}|}{|\{relevant\ tasks\}|} \quad (2)$$

$$F = \frac{2 \cdot P \cdot R}{P + R} \quad (3)$$

$$CC = |\{\text{call graph edges}\}| - |\{\text{call graph nodes}\}| + 2 \cdot |\{\text{connected components}\}| \quad (4)$$

4.2 Execution Procedure

The case study is executed in a finite set of steps, which are partially supported by the REAGENT tool, a plug-in to the INGENIAS Development Kit (IDK) to retrieve MAS models from JADE code.

1. Source code of 19 MAS provided as examples in textbook [2] are considered. These examples are: *base64, behaviours, bookTrading, content, hello, inprocess, jadeJessProtege, jess, messaging, mobile, O2AInterface, ontology, party, pingAgent, protocols, service, thanksAgent, topic and yellowPages*. Benchmarks are manually analyzed to obtain a reference agent model to be used as the *gold standard*.
2. REAGENT v1.0 is applied to each benchmark obtaining an agent model that is graphically visualized with the IDK tool. REAGENT was executed in a computer with a 2.66 GHz dual processor and 4.0 GB RAM.
3. The first sketches of agent models are then analyzed and compared wit*h the gold standard* to obtain base metrics for computing precision and recall (e.g., the number of retrieved non-relevant elements or non-retrieved relevant elements).
4. Having computed all the mentioned measures, collected data are analyzed and interpreted to draw conclusions in order to answer the research questions.

4.3 Data Collection

Table 2 summarizes all the measured values collected during the execution of study for each system. Table 2 shows for each (i) mentioned benchmark, (ii) the number of Java files, (iii) the number of lines of source code, (iv) the cyclomatic complexity average, and (v) the execution time in milliseconds. For each respective agent model, Table 2 provides the number of different INGENIAS elements such as (vi) agents,

Table 2. Data collected during the case study execution

MAS ID	# Files	LOC	C. Complexity	Exe. Time (ms)	# Agents	# Tasks	# Goals	# WFResponsable	# GTPursues	# Rt.	# Rt. Rv.	# Rt. NRv.	# NRt. Rv.	Recall	Precision	F-Measure
1	3	292	3.0	2321	2	0	2	0	1	5	4	1	1	0.80	0.80	0.80
2	4	370	1.5	2858	4	12	0	12	0	28	20	8	0	1.00	0.71	0.83
3	3	456	3.3	2779	2	4	1	4	2	13	13	0	0	1.00	1.00	1.00
4	41	3222	1.3	3027	4	10	0	10	0	24	15	9	10	0.60	0.63	0.61
5	1	44	1.0	2149	1	0	0	0	0	1	1	0	2	0.33	1.00	0.50
6	2	346	2.2	2435	1	1	0	1	0	3	3	0	0	1.00	1.00	1.00
7	7	154	1.7	2080	1	0	0	0	0	1	1	0	3	0.25	1.00	0.40
8	4	677	3.8	2320	1	1	0	1	0	3	3	0	0	1.00	1.00	1.00
9	4	245	2.4	2577	4	3	0	3	0	10	10	0	0	1.00	1.00	1.00
10	6	938	1.8	2455	1	3	0	3	0	7	7	0	0	1.00	1.00	1.00
11	4	185	1.4	2308	1	0	0	0	0	1	1	0	0	1.00	1.00	1.00
12	15	1390	1.7	2764	4	6	0	6	0	16	12	4	2	0.86	0.75	0.80
13	3	994	2.3	2273	2	0	0	0	0	2	2	0	8	0.20	1.00	0.33
14	1	106	2.7	2214	0	1	1	0	0	2	2	0	3	0.40	1.00	0.57
15	5	572	2.9	2671	5	5	0	5	0	15	15	0	0	1.00	1.00	1.00
16	1	62	1.7	1895	0	0	0	0	0	0	0	0	0	Nan	NaN	NaN
17	1	218	9.0	2368	1	1	0	1	0	3	3	0	0	1.00	1.00	1.00
18	2	123	2.5	2278	2	0	0	0	0	2	2	0	4	0.33	1.00	0.50
19	4	347	4.3	2521	3	1	1	1	1	7	7	0	2	0.78	1.00	0.88
Total	111	10741	-	46293	39	48	5	47	4	143	121	22	35	-	-	-
Mean	6	565	2.7	2436	2	3	0	2	0	8	6	1	2	0.75	0.94	0.79
S. Des.	9	736	1.8	287	2	4	1	4	1	8	6	3	3	0.31	0.12	0.24

(vii) tasks, and (viii) goals, as well as relationships such as (ix) *WFResponsible* between agents and tasks, and (x) *GTPursues* between agents and goals. After that, the evaluation of base measures to compute precision and recall are also provided such as the number of (xi) total retrieved elements, (xii) retrieved relevant, (xiii) retrieved non-relevant, and (xiv) non-retrieved relevant elements. Finally, Table 2 provides (xv) precision, (xvi) recall, and (xvii) F-measure.

4.4 Analysis and Interpretation

After the execution of the study, collected data is analyzed to answer research questions. Regarding the effectiveness question (Q1) the analysis of precision and recall is shown in Fig. 2. The recall mean is 0.75 while the precision mean is 0.94. Firstly, a higher precision shows that almost all the retrieved elements in agent models correspond to expected elements. Secondly, a lower recall indicates that there are various actual elements that were not retrieved. However, recall distribution has a high standard deviation, which means that the mentioned problem happens only in some benchmarks (see cases 5, 7, 13, 14 and 18 in Table 2). Anyway, the F-measure is 0.79 on average. This F-measure value is above most reference values of similar works from the literature [9, 17], which define 0.6 as a threshold of applicability.

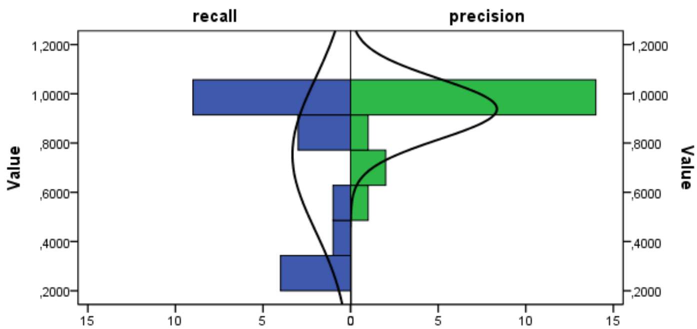


Fig. 2. Comparison of the recall and precision distribution

Despite precision and recall proved to be appropriate, the effect of complexity to these values is also analyzed by means of the anova test. This test analyzes the variance of various sub-samples with respect to a factor. Thus, the null hypothesis is H0: $\mu_1 = \mu_2 = \mu_n$, while the alternative hypothesis means that there is a significant difference between the means of sub-samples, i.e., H1: $\mu_1 \neq \mu_2 \neq \mu_n$. In this study the factor of the anova test is complexity, which was differentiated between low, medium and high regarding percentiles 1/3 and 2/3. Table 3 provides the results of the three anova tests for recall, precision and F-measure. The null hypotheses cannot be rejected in all the cases.

Despite the anova test result is not significant, the effect size of each distribution can be analyzed. Recall improves for high complexity (with a negative sign) and is almost the same for medium complexity. Similarly, precision improves for both medium and high complexity. In general, F-measure follows the same trend. These results imply that complex source code provides better results since there are less missing agent elements and relationships than in toy agent programs with a low cyclomatic complexity.

In addition to Q1, the efficiency of this approach is analyzed regarding Q2. The scalability of REAGENT is tested by a linear regression model (see Fig. 3), which considers the execution time for each benchmark as a dependent variable, and the size of the agent model as the independent variable. Fig. 3 shows the regression line ($y = 31.963 \cdot x - 2195.9$), which presents a positive linear relationship with R^2=0.8244. The correlation coefficient R^2 (between -1 and 1) is the degree to which the real values of the dependent variable are close to the predicted values. The R^2 value obtained is high, and very close to 1, thus the proposed linear regression model is suitable for explaining the data obtained in this study, i.e., there is no quadratic or exponential relationship between the clustering time and the size. The increase in time for larger MAS will consequently be linear, and this time may be assumable. The conclusion is that Q2 can be answered positively.

Table 3. ANOVA Test results for effectiveness measures

Measure	Complexity	Quadratic Mean	F-value	Effect Size	p-value
	Low			0.472	
Recall	Medium	0.244	2.331	0.084	0.129
	High			-0.848	
	Low			0.519	
Precision	Medium	0.120	2.2267	-0.373	0.136
	High			-0.431	
	Low			0.591	
F-Measure	Medium	0.226	3.262	0.035	0.065
	High			-0.931	

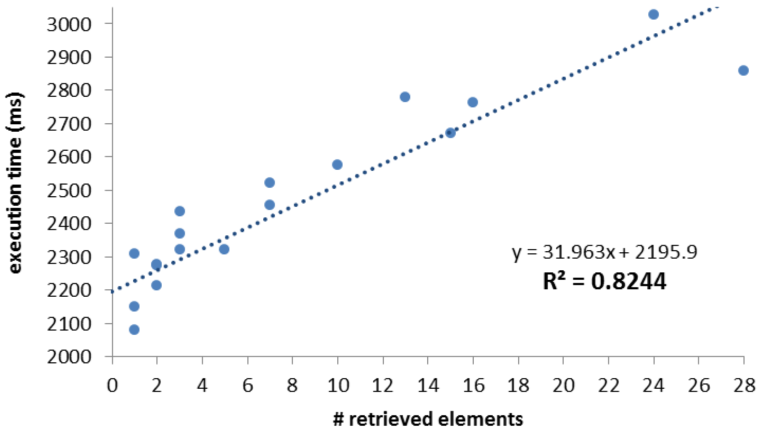


Fig. 3. Linear regression model for the agent model size against execution time

5 Conclusions and Future Work

This paper has presented REAGENT, a reverse engineering technique for retrieving agent design models from legacy Jade source code. The design models are represented according to the INGENIAS methodology. The abstraction of source code toward design models is made following INGENIAS for several reasons: (i) it has been validated in real applications; (ii) it provides useful tools supporting the analysis, design and code generation of based-agent software, (iii) it uses a model-driven approach that facilitates the independence from the implementation platform, and finally (iv) it is not oriented towards a particular agent platform.

A case study with 19 multi-agent systems has been conducted to empirically demonstrate the applicability of REAGENT. The results of the study indicate that REAGENT is able to retrieve accurate and complete agent design models from existing source code. However, particular legacy knowledge is still missing due to the characteristic semantic loss common to all the reverse engineering technique. Despite this fact, REAGENT is able to provide a first sketch of design models, which is obtained in a less error-prone and less time-consuming way than manual modeling from scratch.

The future work will focus on the improvement of REAGENT by incorporating further patterns for considering additional INGENIAS models concerning agent interactions, organization and environment, among others. Additionally, other implementation target platforms will be considered. Finally, a case study involving a large industrial MAS is also expected to be conducted in order to obtain strength results.

Acknowledgements. This work was supported by the FPU Spanish Program and the R&D projects MAGO/PEGASO ([TIN2009-13718-C02-01); GEODAS-BC (TIN2012-37493-C03-01); and Social Ambient Assisting Living - Methods (TIN2011-28335-C02-01).

References

1. Adra, S.F., Kiran, M., McMinn, P., Walkinshaw, N.: A multiobjective optimisation approach for the dynamic inference and refinement of agent-based model specifications. In: IEEE Congress on Evolutionary Computation (CEC) (2011)
2. Bellifemine, F.L., Caire, G., Greenwood, D.: Developing Multi-Agent Systems with JADE (Wiley Series in Agent Technology), 300 p. Wiley (2007)
3. Bosse, T., Lam, D.N., Barber, K.S.: Tools for analyzing intelligent agent systems. Web Intelli. and Agent Sys. 6(4), 355–371 (2008)
4. Canfora, G., Di Penta, M.: New Frontiers of Reverse Engineering. In: Future of Software Engineering (FOSE 2007). IEEE Computer Society (2007)
5. Canfora, G., Di Penta, M., Cerulo, L.: Achievements and challenges in software reverse engineering. Commun. ACM 54(4), 142–151 (2011)
6. Chen, F., Yang, H., Guo, H., Xu, B.: Agentification for web services (2004)
7. Fuentes-Fernández, R., Pavón, J., Garijo, F.: A model-driven process for the modernization of component-based systems. Science of Computer Programming 77(3), 247–269 (2012)

8. Hirst, A.J.: Reverse engineering of Soar agents, pp. 72–73 (2000)
9. Lucrédio, D., de M. Fortes, R.P., Whittle, J.: MOOGLE: A Model Search Engine. In: Czarnecki, K., Ober, I., Bruel, J.-M., Uhl, A., Völter, M. (eds.) MoDELS 2008. LNCS, vol. 5301, pp. 296–310. Springer, Heidelberg (2008)
10. Moreno, J., López, L.: Using Techniques Based on Natural Language in the Development Process of Multiagent Systems. In: Corchado, J.M., Rodríguez, S., Llinas, J., Molina, J.M. (eds.) DCAI 2008, vol. 50, pp. 269–273. Springer, Heidelberg (2009)
11. Open Source Initiative, Java 1.5 grammar for JavaCC (2009), `https://javacc.dev.java.net/files/documents/17/3131/Java1.5.zip`
12. Pavon, J., Gomez-Sanz, J.J., Fuentes, R.: The INGENIAS Methodology and Tools. In Agent-Oriented Methodologies, pp. 236–276. IGI Global (2005)
13. Regli, W.C., Mayk, I., Dugan, C.J., Kopena, J.B., Lass, R.N., Modi, P.J., Mongan, W.M., Salvage, J.K., Sultanik, E.A.: Development and specification of a reference model for agent-based systems. Trans. Sys. Man Cyber. Part C 39(5), 572–596 (2009)
14. Runeson, P., Höst, M.: Guidelines for Conducting and Reporting Case Study Research in Software Engineering. Empirical Softw. Eng. 14(2), 131–164 (2009)
15. Sauvage, S.: Design patterns for multiagent systems design. In: Monroy, R., Arroyo-Figueroa, G., Sucar, L.E., Sossa, H. (eds.) MICAI 2004. LNCS (LNAI), vol. 2972, pp. 352–361. Springer, Heidelberg (2004)
16. Warwas, S., Klusch, M.: Making Multiagent System Designs Reusable: A Model-Driven Approach. In: Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, (2), pp. 101–108. IEEE C. S. (2011)
17. Ye, Y., Fischer, G.: Supporting reuse by delivering task-relevant and personalized information. In: 24th Int. Conference on Software Engineering, pp. 513–523. ACM (2002)

An Agent-Based Analyses of F-formations

Kavin Preethi Narasimhan and Graham White

Queen Mary University of London,
Cognitive Science Research Group,
School of Electronic Engineering and Computer Science, London, E14NS
{k.p.narasimhan,graham}@eecs.qmul.ac.uk

Abstract. The spatial-orientational organization of face-to-face conversational encounters can be described in terms of the sociological notion of F-formations. Ethnomethodological studies of human interaction indicate several geometrical constraints that influence the positional orientations assumed by participants of F-formations. By incorporating some of these constraints, a computational system has been developed that enables agents to dynamically plan their spatial whereabouts in a virtual environment. This paper presents the system and analyses its impact in leading to the emergence of F-formation like spatial configurations in a two-dimensional multi-agent environment. Analyses are based on how experimental subjects perceived the groupings of agents in the virtual environment.

Keywords: F-formations, two-dimensional, multi-agent simulations, social interaction.

1 Introduction

There is growing demand for characters residing in virtual environments to participate in free-standing face-to-face conversations. For example, Second Life and World of Warcraft (WoW) are both game environments in which avatars representing human players participate in virtual face-to-face conversations. It should be noted here that conversations is not the only activity in Second Life or WoW; there are other game related tasks that take priority. In such cases, it is often expected that the virtual characters should by themselves – wholly or partially – be capable of handling some of the conversation related activities, while letting human players handle the game related tasks. Drawing inspiration from sociological theories of human interaction to create computational systems continues to be one of the popular ways of endowing virtual characters with verbal and non-verbal interactional capabilities.

The contributions of this paper are inspired by the sociological notion of an F-formation, which denotes the spatial-orientational configuration that emerges when two or more people (in proximity) orient all or parts of their bodies towards one another during the course of a face-to-face encounter [5]. People generally organize themselves in F-formations so they can have a platform for their collaborative activity. In doing so, there are some practical geometrical constraints

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 239–250, 2013.

that influence the positional orientations assumed by members participating in F-formations. For example, people must stand facing one another within reasonable distances in order to have full visual and auditory access to each other's verbal and non-verbal conversational moves.

This paper will take an approach to simulating F-formations by means of a rule-based system: authors thus adopt the paradigm of statistical mechanics, whereby macro-scale phenomena are derived from behaviour governed solely by micro-scale interactions. In the present case, the macro-scale phenomena are described in Kendon's phenomenology of conversational interaction: attempts will be made to generate F-formations by incorporating appropriate rules that govern the spatial behaviour of individual agents in a hypothesised conversational situation. Rules will be incremented stage by stage, in the hope of generating successively more realistic spatial social behaviour.

The first stage is level 0, which is the most basic implementation in which agents just move and stop at random. This level is a baseline for comparing all other spatial behavioural rules that will be incorporated into the system. The next stage is level 1, which can be hypothesised to apply to human interaction in groups, regardless of whether they are in a conversational situation or not: rules at this level will basically enforce spacing constraints and forbid agents to collide with each other. So then, outcomes of level 0 implementation is examined first, followed by the outcomes of level 1 implementation. Much to the authors' surprise, it was found that quite realistic F-formation like configurations could be generated with only the minimal addition of spatial rules in level 1: the reason for this appears to be that, if agents stop just before they are about to collide, the chances are that they will, at least approximately, be facing towards each other. It therefore appears that adding more sophisticated spatial behavioural rules to the system might yield a better simulation of F-formations, but there is already quite reasonable F-formations with only the minimal rules implemented in level 1.

Discussions here refer to "more realistic" or "better" simulation of F-formations. This, of course, demanded a way of evaluating the outcomes of the simulations so as to be able to make such judgements. Feedback gathered from experimental subjects have been used to evaluate the outcomes of level 0 and level 1 implementations, and to judge how realistic the resulting F-formations are.

2 Related Work

Several approaches have been used to simulate F-formation like spatial configurations in virtual spaces. One approach involves dealing with the spatial-orientational arrangement of agent clusters at a global level. But F-formations aren't a global phenomena – every F-formation is locally organized and managed by people directly involved in it. Therefore, modelling F-formations from a global perspective, like in [15], wouldn't be an ideal solution.

On the other hand, the *social force model* proposed in [4] emphasizes simulating the spatial arrangement of agents on a local scale. [4] uses a rule-based approach to deal with the positioning and re-positioning of every individual agent with a view to sustaining a net force field between agents. Inspired by [4], the rule-based approach adopted by [15] was hugely modified in [8] – wherein rules based on proximity-related data acquired from each individual avatar is used to define a social force field, which in turn guides the active steering behaviour of avatar groups. [8] advocates that this approach is capable of simulating naturalistic F-formations by dealing with *fluctuations* – referring to the spatial-orientational adjustments of a conversational cluster during momentary interventions. The fluctuation concept dealt in [8] was later enhanced in [9] and [10] for enabling avatars to make movement decisions based on an improved set of territorial constraints. A further enhancement in terms of rendering realistic fluid motion was later proposed in [11]. The social force model is, in comparison with the proposed statistical mechanical simulation, something like a mean field approximation. Although it is not entirely global, it suffers from the defect that a mean field approximation is precisely that – an approximation – and the nature of the field remains somewhat questionable. However, the approach followed in this paper is capable of yielding results using only rules which apply to the spatial behaviour of individual agents.

In all these cases, there has also been no direct experiential analysis of how the respective models resulted in spatial configurations that could be perceived as F-formations by real people. It seems unlikely that using post-experiment questionnaires alone can yield a suitable measure of F-formations. Alternate methods like the one used in [2] is required. In [2], authors used a software bot to gather instantaneous information regarding the spatial layouts emerging in Second Life using their computational model. In simple terms, the methodology records information regarding what people experience at the very same instance they experience it. Analysing information gathered using such methods can yield direct measures of the system's impact in leading to the intended outcomes. Though not the same as [2]'s approach, the experimental methods used in this paper will also record participants' instantaneous feedback concerning the spatial-orientational configurations of agents.

Literature surveyed so far concerned modelling the spatial dynamics of smaller groups; there has also been a lot of progress in simulating some of the finer, low-level coordination actions for larger groups of agents – for example [12,7,14]. These models have been quite efficient in simulating the behavioural aspects of larger groups of agents like leader following, flocking, queuing, formation-keeping etc. But aspects like these are a lot different from the dynamics of smaller groups. Nevertheless, some of the concepts from [13,14] have inspired the proposed system in terms of simulating the movement and positioning of agents along an X-Y plane using steering vectors. All this said, it should also be noted that, although agents' motion resulting from some of the path planning algorithms may incidentally lead to spatial configurations that resemble F-formations, they are fundamentally different from the ones discussed here – those that exclusively focus on modelling the spatial dynamics of small groups.

3 Our Approach

3.1 Underlying Sociological Theories

Kendon's (1990) postulates about the features of F-formations forms the theoretical basis of this paper. Reiterating the definition stated in section 1, an F-formation refers to the spatial-orientational organization of face-to-face conversational encounters. Spatial arrangements denoted in figure 1 (a–d) are all examples of F-formations[1]. The centrally enclosed space resulting from any such arrangement is referred to as the o-space (see figure 1). The idea behind the formation of an o-space is as follows. Every individual has a transactional segment – a hypothetical circular sector extending outwards from the midriff of a person's body covering $\approx 30 degrees$ to the left and to the right – that he uses to carry out any activity [5,6]. When alone, an individual constantly endeavours to protect his transactional segment from intrusions. But during collaborative activities like face-to-face conversations, people position themselves such that their individual transactional segments overlap to create an o-space. This o-space offers individuals direct, exclusive and unobstructed access to the verbal and non-verbal utterances of one another. The spatial layout of an F-formation is the outcome of its members' attempts to successfully establish and sustain an o-space, i.e., people will assume spatial configurations that offers them a substantive o-space for their ongoing collaborative activity.

F-formations can be found everywhere (in the road, a party hall, at the corridor etc.), and can assume various shapes (circle, rectangle, triangle, line etc.). The shape assumed by an F-formation is determined by the space between its participants and each of their individual orientations. This is where Hall's (1966) *Proxemics* and his theory about distances in man, both of which are tightly interleaved to the spatial-orientational notion of F-formations, come to play. According to [3], *social distance* is ideal for social interactions among acquaintances (i.e., assuming no personal relationship between interlocutors). Extending outwards from an individual's frontal frame, the social distance covers the area between 4 feet and 12 feet. Within this spatial zone, Hall (1966) determined that a full figure of the human body along with a good deal of space around it is encompassed in a 60-degree glance, which makes it ideal for face-to-face conversations.

3.2 Significance of Individual Positional Orientations

Positional orientations assumed by individuals participating in F-formations serve significant purposes: (1) helps individuals to demonstrate their withness in an F-formation, (2) offers people good stance to optimize their interactions with one another, and (3) contributes in delineating a distinct physical boundary for an F-formation. Even slight variations in individual positional orientations

[1] Figures 1(a–d) are replications of some of the F-formations identified in Kendon's (1990) book chapter.

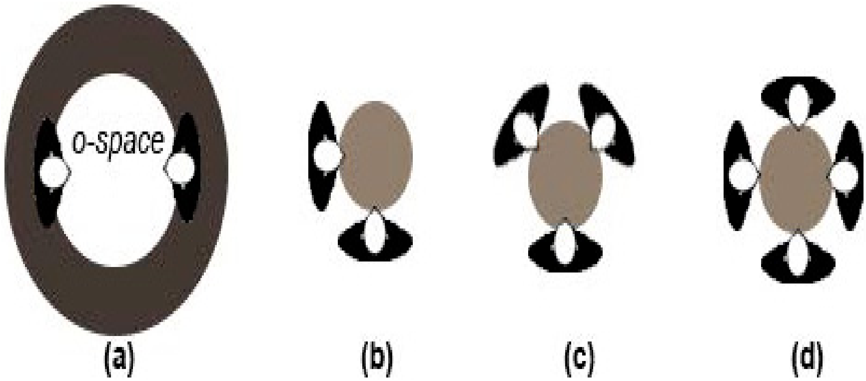


Fig. 1. Examples of F-formations: (a) A vis-a-vis arrangement of two people along with the o-space (overlap of individual transactional segments) (b) An L-shaped arrangement of two people, (c) A triangular arrangement of three people, and (d) A circular arrangement of four people

is bound to impact an interactional occasion in various ways. For example, although an o-space means overlapping individual transactional segments, as a social norm there is an extent of *personal* or *intimate* space immediately surrounding an individual's body – in which physical intrusions are often not encouraged [5]. So when personal spaces are physically intruded, it can either be deemed as a deliberate violation, or that a collaborative activity demands it (e.g., hugging). Considering these aspects, it is possible to hypothesise that: implementing a system of dynamic agent movement and positioning in two different levels – one with rules governing individual positional orientation behaviour and the other without these rules – can yield different simulations of F-formations. And considering the significance of individual positional orientations, it is also likely that the level that includes appropriate spatial behavioural rules will yield a better simulation of F-formations.

3.3 The System – K-Space

Kendon (1990) used a bird's eye view for his analysis of F-formations. Resulting images resembled the series of picture maps denoted in figure 1 (a–d). In order to analyse the implications of the spatial configurations assumed by agents residing in a 2D virtual environment, this paper adopts an approach similar to Kendon's style of analysis. A visualisation platform called K-space was built for this purpose. It is a two-dimensional, multi-agent simulation platform in which agents resemble Kendon's (1990) blob-like representation of humans (see for example, figure 1). Numbers ranging between 0 - 19 are added to the blob-like images for denoting the twenty different agents. The movement and positioning of agents in K-space is based on an extension of the steering behaviour concept introduced in [14]. An initial velocity and position is assigned to each of the twenty agents, which is then dynamically updated as the simulation progresses. Agents move only in the forward direction – denoted by the pointed nose of their blob-like bodies. Agents change their direction of motion either when they have to do so (for reasons explained in section 3.5), or if they reach the boundaries of the simulation window.

3.4 Level 0

When participating in F-formations, people stand facing (more or less) one another within close spatial ranges. Here, quantification of the "close" spatial range isn't possible, nevertheless it is often within the upper limit of an individual's social distance zone and outside his personal space (see sections 3.1 and 3.2). For example, if A and B are talking in an F-formation, A stands outside of B's personal space, but within the upper limit of B's social distance. Apart from some minor temporary fluctuations, people sustain their positional orientation in this intermediate zone for as long as they continue to be a member of an F-formation. This prompts a baseline for the proposed analysis wherein agents move and stop randomly – without obeying any spacing or orientation restrictions. The level 0 implementation serves this purpose; it is the most basic level in which agents just move and stop.

When the system runs in level 0, agents start moving in arbitrary directions – from randomly chosen initial positions and velocities. As the simulation progresses, agents make arbitrary decisions about stopping. After stopping for a prefixed duration, an agent starts to move again, either in the same direction or in a different direction. Figure 2 shows an image generated by running the simulation in level 0. As level 0 rules do not place explicit constraints on the positional orientations of individual agents, there may be instances where agents overlap or cross one another while in motion or stopped (see for example, bottom left of figure 2). That said, as visible in other areas of figure 2, incidentally, agents may also be reasonably spaced and facing one another.

3.5 Level 1

Given the level 0 baseline, the next step is incorporating behavioural rules that govern agents' individual positional orientations, i.e., the level 1 implementation. Theoretical inspiration for these rules are derived from [3] and [5]. According to [3], the social distance zone covers the area between 4 feet and 12 feet extending outwards from an individual's frontal body frame. According to [5], an individual's transactional segment covers $\approx 30 degrees$ to the left and $\approx 30 degrees$ to the right (center being the line extending outwards from the midriff). Together, the two concepts – translated into virtual units of course – are used to define a hypothetical transactional segment for every agent in K-space. Here, a transactional segment is a circular sector whose radius is the virtual equivalent of the upper limit of social distance, and Θ is 60 degrees (i.e., 30 degrees to the left and right of the line extending outwards from an agent's nose).

When the system runs in level 1, agents constantly check if their individual transactional segments – computed based on their spatio-temporal positional orientations – overlap. If there is an overlap, an agent will then (arbitrarily) choose one of the following two actions:

1. *Stop before reaching the lower limit of social distance:* In this case, agents stop anywhere before reaching the lower limit of social distance – measured from the instantaneous position of the other agent; and then stay put in the same positional orientation for some time. After a prefixed duration, agents turn to face a different direction, and then start moving in the new direction.
2. *Turn away before the lower limit of social distance:* In this case, before reaching the lower limit of social distance (calculated in the same way as before) – agents turn away to face a different direction, and then start moving in the new direction. Doing this will eventually cancel the overlap of agents' individual transactional segments because they start moving in different directions. This can be hypothesised to instances where people avoid engagement in conversational encounters by altering their motion paths – happens when passers-by are strangers who wouldn't possibly engage in conversations.

Figure 3 shows an image generated by running the system using level 1 rules. As can be seen, the spatial distribution of agent clusters seen in figure 3 is a bit different from the one seen in figure 2. It should be noted here that except for the addition of the newer rules in level 1, implementation-wise there is nothing different between level 0 and level 1. In so doing, the outcomes of levels 0 and 1 can be compared in terms of the newer rules alone.

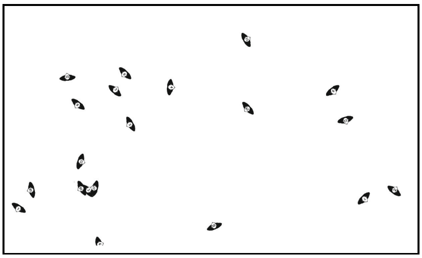


Fig. 2. An instantaneous distribution of 20 agents in the level 0 implementation

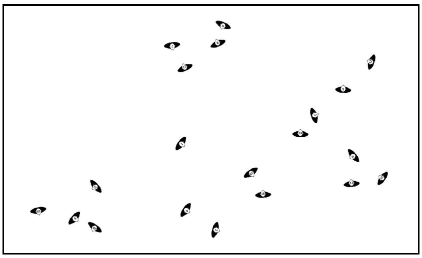


Fig. 3. An instantaneous distribution of 20 agents in the level 1 implementation

4 Experiments

Outcomes from the K-space simulation were subject to user evaluation in two different formats: an image analysis task and a video analysis task. Both the tasks were performed by each of the seven participants who took part in the study. Participants performed the tasks separately at different times and on different days, and took between 60 to 75 minutes to complete both the tasks together.

4.1 Task 1: Image Analysis Task

There were two simulation runs of five minutes length – one each for the level 0 and level 1 implementations. In those five minutes, a static image of the K-space scene showing the instantaneous distribution of agents was saved every time an agent stopped. From all the images that were saved, twenty images (i.e., ten from level 0 and ten from level 1) were randomly chosen to form one set; seven such sets were formed – one for each participant. During the actual test, participants were given the images, asked to study them for some time, and then do the following:

1. Circle around the group of agents (using a green coloured pen) whom they thought might be engaged in a conversation.
2. Describe in written, right next to the circle drawn in step 1, about the kind of conversations that agents were engaged in within those groups.
3. Circle any clearly inappropriate arrangements of agents (using a red coloured pen) and quote the reason for marking them as inappropriate.

While doing the task, participants were neither aware of the level 0 versus level 1 differences, nor of the system itself. So when evaluating the images, participants were oblivious to which level the images belonged to (or even to the concept of levels). The only information on the images shown to participants was a number code that experimenters' used for identification. Evaluation corresponding to the image shown in figure 2 is shown in figure 4, while the evaluation corresponding to the image shown in figure 3 is shown in figure 5.

Fig. 4. An experimental subject's evaluation of the image shown in figure 2

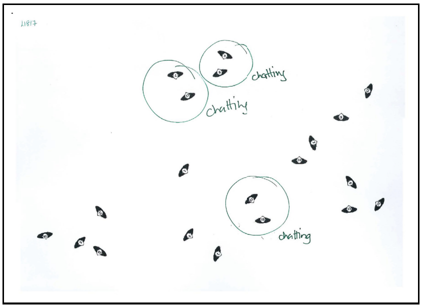


Fig. 5. An experimental subject's evaluation of the image shown in figure 3

4.2 Task 2: Video Analysis Task

Like the image-analysis task, there were two simulation runs of five minutes length – one each for the level 0 and level 1 implementations. Each of them was captured using a screen recording software, and then post-processed to become an interactive video – for allowing experimental subjects to place green and red coloured markers on to the video. Subjects were each shown two videos (one from level 0 and one from level 1). They were asked to track all through (both) the video(s) an agent of their choice, and whenever they thought that agent engaged in conversations, they had to place start (i.e., left mouse button click) and stop (i.e., middle mouse button click) markers next to that agent. Like in the image analysis task, even here participants were not provided with any information regarding the implementation of the videos or the purpose of doing the task. After completing the task, participants were asked to describe their thoughts about agents' activities in each of the two videos they saw.

5 Results and Discussion

For the image analysis task, participants evaluated a total number of 117 images: 59 in level 0 and 58 in level 1. Each image consisted some or no markings of conversational groups and inappropriate arrangements (see, for sample, figures 4 and 5). For the video analysis task, participants evaluated a total number of 14 videos: two each – one corresponding to level 0, and the other corresponding to level 1.

Comparison of the Number of F-Formations: By using the data acquired from the image analysis task, the total number of F-formations that participants identified in level 0 was compared with the total number of F-formations they identified in level 1[2]. Analysing the images showed that the number of F-formations identified in level 1 exceeded the ones that were identified in level 0. With level 1 rules, a total of 183 F-formations were identified; whereas in level 0, a total of 75 F-formations were only identified. The Wilcoxon Signed-Rank Test returned a significance score of 0.018 – suggesting that simulating agent movement and positioning by incorporating level 1 rules significantly enhances the outcome of F-formations. Data acquired from the video analysis task also favoured the same outcome: 28 instances of engagement in F-formations were identified with level 1 rules; whereas only 14 instances of engagement in F-formations were identified in level 0.

Comparison of the Number of Inappropriate Spatial Configurations: Data acquired from the image analysis task was also used to compare the number of inappropriate arrangements of agents identified in the level 0 and level 1 implementations respectively. Analysing the images showed that the number of inappropriate arrangements of agents identified in level 0 significantly exceeded the ones identified in level 1. In level 0, a total of 67 inappropriate arrangements were identified; whereas in level 1, only 25 inappropriate arrangements were identified. The Wilcoxon Signed-Rank Test returned a significance score of 0.018 suggesting that the absence of level 1 rules significantly impedes the emergence of F-formations in level 0.

Comments About Agents' Activities: A comparison was made on the comments (regarding agents' activities in the videos) that participants provided at

[2] Since F-formation is a jargon for people without relevant knowledge, participants were told to identify conversational groups.

the end of the video analysis task. Since agents behave slightly differently in level 0 and level 1, it was anticipated that participants will comment differently for the level 0 video and the level 1 video. For the level 0 video, participants made comments like "Milling around in an open space, like a college quad", "Wandering aimlessly, sometimes stopping randomly, sometimes stopping to interact", "They seemed to move a bit random", "Walking around in an airline terminal in a confused and erratic way". These comments hint that participants may have sensed some sort of randomness in the level 0 video. On the other hand, for the level 1 video, participants made comments like "Busy City Plaza? People rushing about. A station concourse?", "The different speeds [of turning away] made it look more human.", "They seemed more intentional in their movements. The fact that they swing around and made decisive turns made it seem they had "intentions"...Their formation seemed more natural", and "Agent 14 was avoiding talking to people at a public park or train station square."[3]. Even plainly from a descriptive point of view, these comments suggest that participants may have attributed "conversational purposefulness" to agent motion with level 1 rules unlike the "randomness" associated to level 0.

6 Conclusion and Future Work

There are two contributions in this paper. Firstly, a rule-based system to simulate the movement and positioning of blob-like agents in a 2D virtual environment with (in level 1) and without (in level 0) incorporating positional-orientation constraints. Secondly, the K-space system along with the video analysis and image analysis tasks – to gather people's instantaneous feedback regarding the spatial-orientational configurations of agents. The K-space system closely resembles the methodology adopted in [5] to study about F-formations. To the authors' best knowledge, such an approach has not been used before to relate to and confirm the propositions in [5] for agents residing in virtual environments. F-formations emerging in real worlds are principally different from the ones emerging in virtual spaces, therefore analyses of the kind reported in this paper is an essential preparatory step for building computational models that can simulate human-like F-formations.

There are also some limitations to the proposed work. Firstly, not all factors influencing F-formations have been dealt with in this paper. People's choice for spacing and orientation within F-formations can be influenced by multiple factors like: the total number of people involved – their relationship with one another, cultural background, gender, social status etc.; nature of the collaborative activity (or conversation) for which an F-formation is established; physical environment within which F-formations function etc. Each of these factors play a role in influencing the spatial-orientational organization of F-formations. Therefore, a holistic approach to simulate F-formations in virtual environments should deal with the aforementioned factors alongside permitting the exchange of verbal utterances. Nevertheless as complex as that seems to implement and test within

[3] This comment was stated by the participant who tracked agent no. 14.

the scope of one paper, analyses here is restricted to the spatial-orientational arrangements alone.

Secondly, using blob-like agents made it impossible to analyse the impact of features like gaze (or head orientation) and gestures in contributing to the emergence of F-formations in virtual space. In real life, people make optimal use of their body parts – head, body, hands and legs – while participating in F-formations. Therefore, humanoid agents that are capable of using their body parts when establishing spatial-orientational relationships, are also required to confirm the hypothesis proposed in this paper. Figure 6 shows authors' WIP on this front implemented using the Smartbody platform[4].

To sum up, this paper has shown that adding minimal spatial behavioural rules to the dynamic movement and positioning of individual agents can significantly enhance the outcome of F-formations. That said additional rules concerning the spatio-temporal arrangement of agents will need to be added to the rule-based K-space system for validating its overall efficiency in simulating F-formations.

Fig. 6. Simulation with humanoid agents

References

1. Ciolek, T., Kendon, A.: Environment and the spatial arrangement of conversational encounters. Sociological Inquiry 50(3-4), 237–271 (1980)
2. Friedman, D., Steed, A., Slater, M.: Spatial social behavior in second life. In: Pelachaud, C., Martin, J.-C., André, E., Chollet, G., Karpouzis, K., Pelé, D. (eds.) IVA 2007. LNCS (LNAI), vol. 4722, pp. 252–263. Springer, Heidelberg (2007)
3. Hall, E.: Distances in man: The hidden dimension, pp. 113–129. Double Day, Garden City (1966)
4. Jan, D., Traum, D.: Dynamic movement and positioning of embodied agents in multiparty conversations. In: Proceedings of the Workshop on Embodied Language Processing, pp. 59–66. Association for Computational Linguistics (2007)

[4] http://smartbody.ict.usc.edu/

5. Kendon, A.: Conducting interaction Patterns of behaviour in focused encounters, pp. 209–237 (1990)
6. Kendon, A.: Spacing and orientation in co-present interaction. In: Esposito, A., Campbell, N., Vogel, C., Hussain, A., Nijholt, A. (eds.) COST 2102 Int. Training School. LNCS, vol. 5967, pp. 1–15. Springer, Heidelberg (2010)
7. O'sullivan, C., Cassell, J., Vilhjalmsson, H., Dobbyn, S., Peters, C., Leeson, W., Giang, T., Dingliana, J.: Crowd and group simulation with levels of detail for geometry, motion and behavior. In: Third Irish Workshop on Computer Graphics, vol. 1, p. 4 (2002)
8. Pedica, C., Vilhjálmsson, H.H.: Social perception and steering for online avatars. In: Prendinger, H., Lester, J.C., Ishizuka, M. (eds.) IVA 2008. LNCS (LNAI), vol. 5208, pp. 104–116. Springer, Heidelberg (2008)
9. Pedica, C., Vilhjálmsson, H.H.: Spontaneous avatar behavior for human territoriality. In: Ruttkay, Z., Kipp, M., Nijholt, A., Vilhjálmsson, H.H. (eds.) IVA 2009. LNCS, vol. 5773, pp. 344–357. Springer, Heidelberg (2009)
10. Pedica, C., Vilhjálmsson, H.: Spontaneous avatar behavior for human territoriality. Applied Artificial Intelligence 24(6), 575–593 (2010)
11. Pedica, C., Vilhjálmsson, H.: Lifelike interactive characters with behavior trees for social territorial intelligence. In: ACM SIGGRAPH 2012 Posters (SIGGRAPH 2012), Article 32. ACM, New York (2012)
12. Silveira, R., Prestes, E., Nedel, L.P.: Managing coherent groups. Comput. Animat. Virtual Worlds 19(3–4), 295–305 (2008)
13. Reynolds, C.: Flocks, herds and schools: A distributed behavioral model. ACM SIGGRAPH Computer Graphics 21, 25–34 (1987)
14. Reynolds, C.W.: Steering behaviors for autonomous characters. In: Proc. of the Game Developers Conference, San Francisco, CA, pp. 763–782. Miller Freeman Game Group, New York (1999)
15. Vilhjalmsson, H., Merchant, C., Samtani, P.: Social puppets: Towards modular social animation for agents and avatars. In: Schuler, D. (ed.) HCII 2007 and OCSC 2007. LNCS, vol. 4564, pp. 192–201. Springer, Heidelberg (2007)

X-CAMPUS: A Proactive Agent for Ubiquitous Services

Hajer Sassi and José Rouillard

NOCE Team. Laboratoire d'Informatique Fondamentale de Lille,
Bat M3 - F-59655 Villeneuve d'Ascq cedex

Abstract. Our research is at the crossroads of two main complementary areas: agents for intelligent systems and human-computer interactions. The prospect of highly natural interfaces, in which interaction between user and computer is mediated by lifelike agents, is currently one of the major steps leading to the acceptance of relevant and practical agents system by people. In this paper we illustrate our approach which takes into account both the evolution of the user interface technology and the emergence of ubiquitous computing in order to propose a proactive adaptive assistance through an agent named X-CAMPUS (eXtensible Conversational Agent for Multichannel Proactive Ubiquitous Services). It aims to assist proactively user in her daily tasks thanks to its ability to perceive the state of the environment and to communicate effectively according to the user's current situation.

Keywords: Intelligent Interfaces, ubiquitous computing, human-computer interaction, proactive assistance, multimodal interfaces, multichannel interfaces.

1 Introduction

Numerical world searches to make technologies more efficient through the development of system able to assist user according to her current situation thanks to the emergence of new devices able to communicate information that described user's environment. Many types of smart applications have recently appeared in order to help user reactively. This kind of applications requires user's intervention in order to satisfy her/his needs or thanks to ubiquitous devices and their capability to provide information which describe user his/her surroundings she/he can be satisfied without any explicit request.

We have decided to develop an approach which can determine user's needs based on contextual information that our agent has collected during the communication with the user.

The name "X-CAMPUS" chosen for our agent summarizes the principal characteristic that we have decided to study in our approach. X-CAMPUS stands for eXtensible Conversational Agent for Multichannel Proactive Ubiquitous Services:

- eXtensible: our system is not limited to only one application domain, but can be extended to many different ones (home automation, e-learnig, e-commerce, entertainment, tourism, etc.) according to the sensors available in order to capture the interaction context.

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 251–262, 2013.

- Conversational: our system is able to dialog with the user in different conversational modes (menu, question/answer), in order to bring more natural communication between users and agents. It collects and provides information across several instant messaging and VOIP services (Gtalk, MSN, IPPI...).
- Agent: in our system, each user is connected to a digital agent. The main engine of our system (called Athena) is able to manage multiple conversations at the same time, and to trigger relevant actions according to a particular context. Some rules are used to decide each (simple or complex) action to perform according to various events (time, profile and location of the user, preferences, etc.).
- Multichannel: in our hyper-connected world, we have decided that X-CAMPUS would be able to communicate with users across multiple channels. We have particularly worked on Internet and phone channels, thus if a user is not connected online (Internet instant messaging), our system tries other possibilities (vocal phone call, SMS, E-mail...) in order to reach the user, proactively.
- Proactive: unlike most of the classical agent systems, mainly conceived on a reactive-based logical (user's request gives agent's response). X-CAMPUS was programmed with the possibility that the digital agent could push information to the user without any explicit solicitation.
- Ubiquitous: X-CAMPUS manages a range of heterogeneous devices in order to collect information and consequently to be able to assist user anywhere, anytime, in a completely ubiquitous and pervasive way.
- Services: our agent perceives user’s environment permanently. Once it detects a change in the state of the user’s surroundings it proceeds to analyze the new situation immediately and to determine if it should communicate information and provide relevant services to the user.

This paper is organized as follows: section 2 describes a walk-through example. In section 3 we detail our approach and we illustrate our agent’s behavior in section 4. Section 5 reports some experimental results. Finally, Section 6 presents our conclusions and future work.

2 Our Approach

The main goal of the use of ubiquitous devices is to ensure access to the state of our surroundings which formed user’s current context. This term appeared in 1992 with Want and al. when they introduced their system named Active Badge Location System which is considered to be one of the first context-aware applications. It aims to build a system for phone calls delivery according to the called person’s localization.

After that many other definitions were succeeded and most of them are responses to the following questions: why? What? Where? When? and how? That will generate an enormous quantity of useful and useless information [2]. Chen and Kotz [3] concluded that existent definitions of context are very general, vague and not clear enough to help its understanding in computing systems. In the following we will give, in chronological order of appearance, a non-exhaustive list of the most relevant definitions of context which take into account explicitly the human-computer interaction.

In 2000, Dey defines context as “any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves” [5]. Salber [15] described context as any information required by the interaction between the user and the application which can be sensed by the application. Miraoui and al. define context as “Any information that trigger a service or change the quality of a service if its value changes” [13]. In our work, to use context effectively we have decided to define context as any information that let our system initiate a conversation or change the way of interaction in order to ensure the continuity with the user.

3 Architecture

Context aware systems can be implemented in different ways according to systems their requirements and their conditions such as the location of sensors (local or remote), the amount of possible users (one user or many), the available resources of the used devices (high-end-PCs or small mobile devices) or the facility of a further extension of the system [1]. Furthermore, the method of context-data acquisition is very important when designing context-aware systems because it predefines the architectural style of the system at least to some extent. Chen presents three different approaches on how to acquire contextual information: direct sensor access, middleware infrastructure and context server [4].

In order to exploit pervasive information around us, we need to follow a context-model that define and store context data. Strang and Linnhoff-Popien summarized the most relevant context modeling approaches, which are based on the data structures used for representing and exchanging contextual information in the respective system [17]. So he classifies context-models in six models which are key-value model, markup scheme models, graphical models, object oriented models, logic based models and ontology based models. For technical reasons we have decided to model our context by using object-oriented techniques which offers to use the full power of object orientation (e.g., encapsulation, reusability, inheritance).

Our architecture is a framework developed on .net technologies based on three main layers. It principal vocation is to develop a digital ubiquitous agent. Once, agent is developed, it will be deployed on a server containing a database of contextual information (based on an object oriented model for context description), supporting various constraints and rules developed in an ad-hoc way. Our system has the capability to manage many users simultaneously; it can communicate with client (applications) through three different protocols which are Microsoft Notification Protocol (MSNP) [10], Extensible Messaging and Presence Protocol (XMPP) [9] and Simple Object Access Protocol (SOAP) [11]. Moreover of these protocols, our approach uses a scripting language based on the mechanism of the Artificial Intelligence Markup Language (AIML) [8] in order to facilitate the development of conversational agent. This language named Athena [12] is composed of several units, the most relevant ones are: the pattern, the template, the label and the scenario. The first unit “pattern” is reserved about what user should say, whereas the second unit "template" is about what agent should response. The third unit "label" is reserved to trigger (initiate) a proactive conversation and the last unit "scenario" is a set of pattern/template.

3.1 Context Manager Layer

To build systems able to act differently according to context awareness, intelligent environment should perceive and control sensors networks regularly through the "context manager layer". This layer should communicate with heterogeneous sources in order to collect information and register them in the database. This layer is based on context provider and context repository. It controls the behavior of sensors and saves new issues values (static, dynamic and temporary information) in context repository.

Static information represents data which remain unchanged during a long period, such as user's name, surname, age, etc., which are communicated by user during the first exchange with X-CAMPUS. However dynamic information changes frequently (examples: user's calendar events, location…). X-CAMPUS communicates with a SOAP client who manages two logical sensors: Google Calendar and Google Latitude. These two dynamic sources of information are perceived continuously in order to ensure a best proactive assistance to the user. The last types of information managed by X-CAMPUS are those strongly related to the time (date and/or hour). It's for example the moment chosen by the users from the forecast weather service notification.

This layer communicates directly with the second layer in order to publish information even before context repository registers information in database for later use. An example of sensors that we used to collect information is a Radio Frequency Identification (RFID) reader accompanied with RFID tags. When RFID reader detects an RFID tag, it firstly determines the user's name in order to salute her and secondly calculates the number of persons at home [16]. To gather user's information (current activity and preferences), we have chosen to ask some questions according to the user's context among an xml file which contains some questions grouped by theme [16]. User can also enter data through a software entity (e.g., Website, Google calendar, Face- book, etc.) and provide access to system which can use this software in order to more help user. This layer distinguishes three types of information: the static information, the temporary information and the dynamic information. Static information remains unchanged during the process of learning (e.g., name, age, etc.). Temporary information can be sometimes changed (e.g., preferences, taste, etc.). However dynamic information changes frequently (e.g., location, mood). All these types of information are stored in a database in order to be used later.

3.2 Anticipate Needs Layer

In our research, we are based on "context manager layer" in order to anticipate user's services. In this layer, we try to exploit stored data context manager by associating a set of adaptive operators. Actually, we distinguish three types of operators [16]:

Conversion operator which adapts the original format in order to associate a meaning manageable by the system. For example: when temperature sensor sends the raw data "2", the conversion operator interprets this value as "it's cold" or "it's hot", according to the real situation of the user. Extract operator which extracts only relevant information. Example: extract just the minute from the current time. Coupling operator used when system should aggregate various and heterogeneous (logical and/or physical) data. Thus we propose a coupling operator which tries to collect many data in order to "understand" non-trivial situations. For example detecting the location of users in a living room requires gathering information from multiple sensors throughout the intelligent home. It should also, in many cases, combine the results of several techniques such as image processing, audio processing, floor-embedded pressure sensors, etc., in order to provide valid information.

3.3 User Interface Layer

In a ubiquitous environment, the behavior of services does not depend just on explicit user interaction but also on environment's perceptions. Combing these two sources of information, system can better respond to user's expectations. Our system has to provide an adaptive way of interaction according to the user's situations.

The "user interface layer" should be able to define the context and choose the best way to interact by selecting the appropriate modalities and channels [16].

4 Illustration and Scenarios

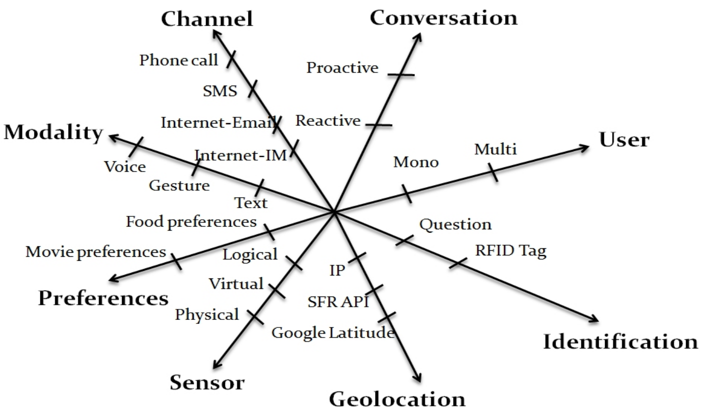


Fig. 1. X-CAMPUS's capabilities

As we can see in Figure 1, X-CAMPUS has many different skills which are represented by the different available axes. Thanks to the use of physical (e.g., RFID tags), virtual (e.g., Google calendar) and logical sensors (e.g., Google calendar combined with RFID tags), our system can interact in both reactive and proactive mode, according to user's needs. X-CAMPUS is also capable to manage some social aspects of the context. For example, events like "Eating at restaurant" registered in the user's agenda will lead to take into account the possible guests (friends, colleagues, etc.) involved in this particular event. Thus, some relatively complex and time consuming tasks can be done directly by a software agent that is in charge to find the better choice (example: the best restaurant) according to various criteria (alimentary preferences, distances and geolocation of each person, etc.). In order to ensure an adaptive interaction, we have decided to work on multi-channel and multi-modal interfaces and we have chosen to use two types of channels which are Internet and phone [14] and three types of modality which are text, gesture and voice.

We have proposed in [16] a scenario talking about favorite TV shows. With this scenario we have demonstrated the capability of our agent to collect relevant information concerning favorite TV categories of various users thanks to logical sensors (RSS with category, date, hour, duration…) which allows X-CAMPUS to identify user's favorite programs among many flows. It was also able to find the best way to communicate this information according to user's situation (at home, at office, alone, etc.)

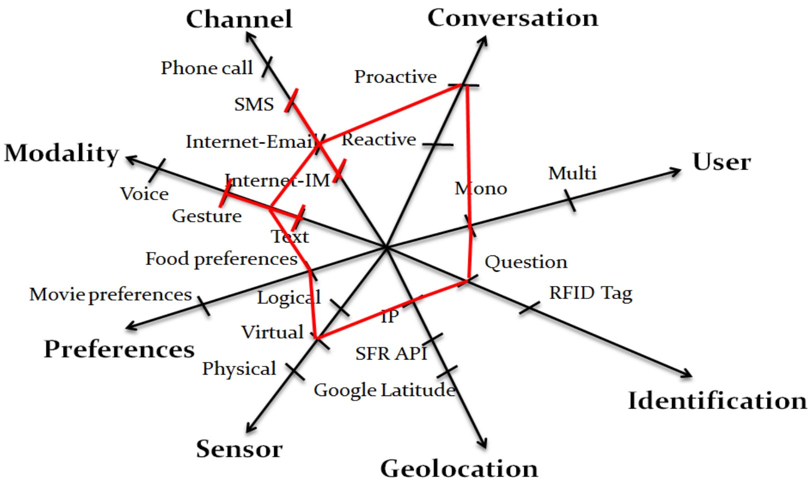


Fig. 2. X-CAMPUS's capabilities used in the favorite dishes

4.1 User's Favorite Dishes

As we see in Figure 2, X-CAMPUS matches all axes by integrating one or more values for each capability. We will give more information about this figure in the section below.

Internet Channel

This scenario is about Mr. Marc's favorite dish. System knows that Mr. Marc likes "potatoes, beef and pizza" and wants to be notified at 11 o'clock AM. The context used to satisfy user's favorite dish is: user's food preferences, user's favorite notification period, user's phone number, user's e-mail and restaurant menus. Every day, our

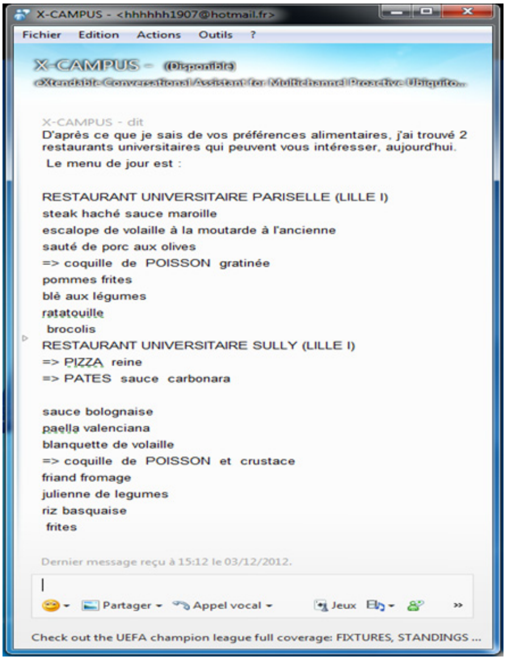


Fig. 3. X-CAMPUS notifies user about her favorite dishes

system checks user's favorite notification period, menus proposed by restaurants and user's preferences dishes. If X-CAMPUS finds a minimum of one restaurant that contains a minimum of one of user's favorite dishes, it calculates the remaining time from the start period of notification and decides to send this information to the "User Interface Layer". Afterward, this latter layer sends a request to the "Context layer manager" in order to determine user's situation. For example, at the office and when user is connected, the system will provide this service using a classical text modality (MI) by sending information which contains the name of the restaurant, its menu, and on upper case user's favorite dishes through the Internet channel (see Figure 3). We can see that X-CAMPUS justifies its decision in order to let users understand easier its behaviors.

As we can see in figure below, X-CAMPUS has proposed two restaurants that their menu contains user's favorite dishes which are "Poisson", "Pizza" and "Pates". User can be disconnected from the Internet and in this case our system should find another way to communicate at the appropriate period/time. So thanks to a second channel that we use in our approach, service can be communicated in time through the phone-channel.

Phone Channel

As we said in previous sections, we tried to provide proactive intelligent interfaces which can associate different types of modalities with different channels. However, when the user is disconnected from the internet network channel, and if the agent has important information to communicate to her, it should find a new way of communication to reach her wherever she is (home, office, outside, etc.). So, as second channel of communication that can be interesting in our work, we have chosen the phone channel, which allows our system to communicate with people when they are disconnected from the internet. This step is very important in our research; it ensures the continuity with the user by sending for example a Short Message Service message (SMS) as illustrated with Figure 4.

Fig. 4. Sending SMS through phone channel when user is disconnected

4.2 Complex Events

People frequently organize social events in order to meet, work, or discuss together. This kind of events are complex to managed (time and energy consuming) and to organized correctly. For example, going to the cinema, eating together, etc. can be some very difficult tasks to manage (for a human) as various and multiple criteria can be involved (time, weather, movie category, restaurant style, preferences of each person...).

Obviously, most of the time, the task in more and more complex as the number of invited is increasing. And sometimes, it becomes just impossible to find a convenient solution for everyone. In ordinary case, the organizer should call all participants or invite them through an electronic application such as Google Calendar or Doodle. As a consequence organizer should control by himself guests' responses and find the best solution which satisfies all participants.

In order to help users in these situations, X-CAMPUS has the capability to manage complex situations (multi-user, multi-modality, multi-channel...), according to user's situations, we have decided to integrate a new virtual sensor named “Google Calendar Sensor”. Its vocation is to read user's Google Calendar and to send new events to our system. Once the system receives new events, it updates guests' profiles.

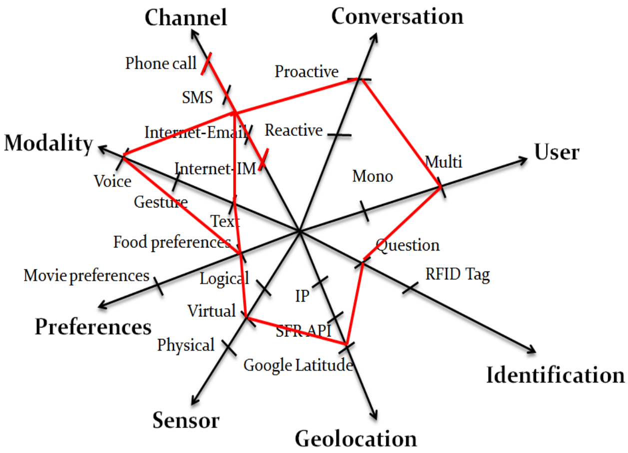


Fig. 5. X-CAMPUS's capabilities used in favorite social dish service

As we can see in Figure 5, X-CAMPUS will initiate a proactive conversation by considering as contextual information: user's mode (multi-user), user's location, user's food preferences, user's state (connected or disconnected), and also the possibility to change the used modality (text, voice) and channel (phone or Internet).

For example, a user (say Marc) decides to use Google Calendar to invite his two colleagues and to organize a meeting in a restaurant (criteria when and where). In this kind of situation, X-CAMPUS is switching from a non-social mode to a social mode. The difference between them is located at the "User Interface Layer". Once system decides to communicate service it checks if Marc has a social event named "Eating at university restaurant". According to system's check, it decides if it should behave in social or non-social mode. In the first case, we find ourselves with "user's favorite dish" whereas in the second case, system should check guests' favorite foods (except current user) thanks to the possibility to access to user's profile from her address mail (the address used to send invitation). Thereafter, it can determine guests' available restaurants more precisely the name of each restaurant that its menu contains at least one of guest's favorite dishes. Now, system knows each guest's available restaurants, so it starts its analysis phase in order to find common restaurants. We worked on two different solutions to propose a restaurant to the organizer: a Boolean method and a more complex one (sorted list). With the first method, if a restaurant does not propose at least one favorite dish for a user, it is not selected. As we can see in Table 1, our system can find no common restaurants, one restaurant, or many restaurants which satisfy guests' needs.

The ideal situation is the case where only one restaurant satisfies all the users' preferences food (see Table 1). In this case it communicates by using a classical text modality through Internet channel or phone channel according to user's state. The message sent by the system contains the name of the chosen restaurant, the menu, the names of the guests and the user's favorite dishes (in uppercase). In a quite simple

Table 1. X-CAMPUS's social behavior for two different days

	First day				Second day			
User / Restaurant	User #1	User #2	User #3	User #4	User #1	User #2	User #3	User #4
Restaurant #1	Pasta	Beef	Pizza	chips	Chiken			Fish
Restaurant #2	Pizza	Potatoes	Beef, pizza	Hambur ger	Potatoes	Beef	Pizza	Pasta
Restaurant #3	Pizza, potatoes	Beef	Pizza	Pasta		Pizza	Fish	

Three common restaurants — One common restaurant

situation, X-CAMPUS is able to check, each day, for each user, the best suitable (university) restaurant to propose to the users according to various personal criteria stored in the user profile, such as alimentary preferences (Pizza, Beef, Pasta...), status (student, teacher...), etc. X-CAMPUS users are daily notified about the best match between their preferences and the different menus proposed by all the restaurants available in their geographical area. This notification is launched at a moment previously chosen. It can be done across an instant messaging tool (MSN, GTalk...), or by E-mail, or by SMS, according to the context of use. In a more complex manner, X-CAMPUS is also able to manage situations where people have planned to eat together: it calculates the more relevant restaurant(s) based upon the preferences of all the users involved in a particular meeting. This second resolution method tries to sort the restaurants, according the users' preferences, and is able to select the best solution, even degraded. Figure 6 and 7 present the logical optimization routine used to calculate the best list of restaurant to choose for a single user or for a group of users. The first example (Figure 6) is relatively simple, because the restaurant A is a solution for three users among five, and is naturally proposed to the organizer, as the best restaurant to choose, in this context.

		USER				
		1	2	3	4	5
RESTAURANT	A	1	1	2	2	3
	B	3	1	0	0	2
	C	1	0	3	1	3
	MAX	3	1	3	2	3
	Position max	2	1	3	1	1
	Choosen restaurant (by user)	B	A	C	A	A
Number of :	A	3				
Number of :	B	1				
Number of :	C	1				
Choosen restaurant for the group:	Position:	1				
	Value:	A				

Fig. 6. X-CAMPUS's decision (first situation)

In the second example (Figure 7), we can see that 4 preferred items of the user 1 (say for instance "chicken", "beef", "salad" and "pizza") are proposed by restaurant A.

In a "mono user" mode, the best suitable restaurant for user 1, 2, 3, 4 and 5 are respectively the restaurants A, C, B, C and B. In a "multi user" mode, our system has to propose the restaurant that maximizes the user constraints (favorite dishes, here). Thus, the restaurant B is proposed to the organizer because two users among five will have four favorite dishes in this restaurant.

		USER				
		1	2	3	4	5
RESTAURANT	A	4	1	2	0	3
	B	3	2	4	2	4
	C	1	3	3	3	3
	MAX	4	3	4	3	4
	Position max	1	3	2	3	2
	Choosen restaurant (by user)	A	C	B	C	B
Number of :	A	1				
Number of :	B	2				
Number of :	C	2				
Choosen restaurant for the group:	Position:	2				
	Value:	B				

Fig. 7. X-CAMPUS's decision in case two (second situation)

If a unique solution is available, then X-CAMPUS notifies directly all the users by indicating the chosen place. But if there is not only one solution, our system is able to trigger a communication with the organizer in order to choose across a conversation, the best contextual criteria to deal with this complex situation. Technically, when X-CAMPUS finds multiple solutions to a given situation (restaurants, movies, etc.) a vocal conversation is initiated with the organizer in order to take a decision. We are using Ippi Messenger [6] and Tropo [7] for this purpose. Ippi Messenger is a Voice Over IP (VOIP) tool, compatible with the Session Initiation Protocol (SIP). Tropo is a powerful yet simple API that adds Voice and SMS support to the programming languages that programmers already know (JavaScipt, PHP, Ruby, Python, Groovy...). In our example, when our system finds more than one restaurants, it decides to place a vocal call to the organizer. During this conversation, some other criteria are proposed to enlarge the contextual information set (geolocation of the participants, for instance).

5 Conclusion and Outlook

In this paper, we have presented our agent named X-CAMPUS which searches to increase the productivity and the welfare of the user situated in intelligent environments. We have illustrated our approach among a proactive multichannel and multimodal service in both simple (mono user) and complex way (social event). In the simple way X-CAMPUS searches to satisfy user according to her favorite dishes by proposing only menus of restaurants which contain user's favorite dishes. After that it analyses user's state in order to ensure the best way of interaction thanks to the notion of multimodality and of multichannel. Whereas in a complex way X-CAMPUS searches to satisfy more than one user simultaneously among two different solutions: a Boolean method and a more complex one (sorted list). Currently, we envisage an evaluation with users by proposing a set of proactive services in order to study users' behavior and our agent capabilities to adapt its behaviors according to user's context.

References

1. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. Int. J. Ad Hoc and Ubiquitous Computing 2(4) (2007)
2. Brézillon, P., Borges, M.R., Pino, J.A., Pomerol, J.C.: Context-Awareness in Group Work: Three Case Studies. In: Proc. of 2004 IFIP Int. Conf. on Decision Support Systems (DSS 2004), Prato, Italie, Juillet (2004)
3. Chen, G., Koltz, D.: A survey of context-aware mobile computing research, Department of Computer Science, Dartmouth College, Technical report (2000)
4. Chen, H.: An Intelligent Broker Architecture for Pervasive Context-Aware systems, PhD Thesis, University of Maryland, Baltimore County (2004)
5. Dey, A.K.: Understanding and using context. Personal and Ubiquitous Computing 5, 4–7 (2001)
6. http://www.ippi.fr/index.php?page=home&lang=44&lang=44
7. https://www.tropo.com
8. http://www.alicebot.org/aiml.html
9. http://xmpp.org/xmpp-protocols/xmpp-extensions/
10. http://msnpiki.msnfanatic.com/index.php/MSN_Protocol_Version_18
11. http://www.w3schools.com/soap/soap_intro.asp
12. http://www.xbrainsoft.com/
13. Miraoui, M., Tadj, C., et al.: A service oriented definition of context for pervasive computing. In: The 16th International Conference on Computing. IEEE Computer Siciety, Mexico city (2007b)
14. Rouillard, J., Tarby, J.-C., Le Pallec, X., Marvie, R.: Facilitating the design of multichannel interfaces for ambient computing. In: The Third International Conferences on Advances in Computer-Human Interactions, ACHI 2010, St. Maarten, Netherlands Antilles, pp. 95–100 (2010)
15. Salber, D.: Context-awareness and multimodality. In: Colloque sur la Multimodalité. IMAG, Grenoble (2000)
16. Sassi, H., Rouillard, J.: Proactive Assistance within Ambient Environment. In: Intelligent Systems and Applications, INTELLI 2012, Chamonix, pp. 60–65 (2012)
17. Strang, T., Linnhoff-Popien, C.: A Context Modeling survey. In: The First International Workshop on Advanced Context Modeling, Reasoning and Management, UbiComp 2004 (2004)

Agents Vote against Falls: The Agent Perspective in EPRs

Sebastian Ahrndt*, Johannes Fähndrich, and Sahin Albayrak

DAI-Labor, Technische Universität Berlin,
Ernst-Reuter-Platz 7,
10587 Berlin, Germany
sebastian.ahrndt@dai-labor.de

Abstract. In this work we present an agent-based fall-risk assessment tool which is self-learning. As part of a mobile electronic patient record (EPR) each patient is represented by its agent which helps to lift the treasure of data offered by combining multiple EPRs in order to reveal personalized health-care. To learn from the data provided by the population under care, we enabled the patient agents to negotiate about possible fall-risk indicators using a distributed information fusion and opinion aggregation technique.

1 Introduction

The prevention of falls is a priority objective of nowadays health-care professionals as falls do not only contract injuries and disabilities, but also lead to substantial economic burdens (0.8% - 1.5% of the yearly German health-care costs [8]). Usually health-care professionals utilize fall-risk assessment tools to conceive the individual fall-risk of the patients and as a consequence to initiate suitable retaliatory actions. Here the increasing dissemination of electronic patient records (EPR) offers new opportunities for a more personalized health-care as EPR contain a treasure of data [6]. Even so, the ubiquitous access to all patient information can overwhelm the professionals identifying relevant fall-risk indicators as the feature space is to huge to conceive for humans. Hence, we introduce a patient agent which observes the whole feature space in order to learn the influences of each feature on the fall-risk of a patient. Further, we enabled the patient agents to negotiate about possible fall-risk indicators using a distributed information fusion and opinion aggregation technique. The negotiation results are then used to adapt the fall-risk assessment tool to the population under care. The focus of this work is to demonstrate the capabilities of the application by providing insights into the approach as well as insights into the simulation accomplished for the evaluation and the final application as part of a mobile EPR developed within the project agnes^{zwei} [2].

* Corresponding author.

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 263–266, 2013.

2 Main Purpose

The environment we address consist of several health-care professionals, each one equipped with a Tablet which is the runtime for the mobile EPR. Every Tablet is represented by an agent node containing several patients, where each patient is represented by its patient agent. The learning process is based on four steps: The occurrence of a fall, the local information fusion, the global opinion aggregation and the notification stage. Whenever a patient falls, the incident is documented in the associated EPR by the health-care professional. For each observed fall-event the patient agent elaborates the evidence for all features in the feature space using Bayesian information fusion [9]. Here, the prior knowledge about all features is linked with the current observation, where the initial knowledge of each agent where received from a literature research we conducted [11]. The received probability distribution can be seen as 'degree of believe' about possible fall-risk indicators [7]. Consequently each patient agent has an own expert opinion on which feature has influenced the fall observed. To receive a meaningful statement the challenge now 'is to pool opinions in a rational way that yields a single probability distribution from which inference can be made' [10]. Here, we first pool all opinions on a single Tablet and only pool the node opinion with the more global opinion of all Tablets if a feature reaches an experimentally established threshold. We decided to introduce this two stages as the communication on one Tablet is cheap and secure. As opinion aggregation technique we applied the 'Supra-Bayesian Pooling Method' [5] as it outperforms other evaluated pooling methods for the specific domain we address (rather small occurrence of events, fast convergence, linear time and space complexity). However, if the threshold of a feature is still exceeded after the global opinion aggregation the elaborated feature is forwarded to the health-care professional which has to decide whether this feature is a fall-risk indicator. For more details about the approach and the evaluation the interested reader is referred to *Ahrndt* and *Fähndrich* [1].

3 Demonstration

The goal of the demonstrator is twofold. To start with, we want to show how the overall system works and how the negotiation process between the single agents and the agent platforms operates. Therefor we will show the simulation environment we setup for the evaluation of the approach at runtime utilizing the ASGARD agent viewer [12] (see Fig. 1.2). We utilized the simulation runs in order to test the applicability and validity of the system and further to evaluate the performance of different distributed information fusion techniques. Here, we will show the progress of the probability distribution during runtime (see Fig. 1.3). To create meaningful results, each patient-agent where equipped with a set of fall-risk indicators received from a literature research [11] and the data model of the EPR developed within the agnes^{zwei} project [2]. To initiate the negotiation we further implemented a simulation agent which represents the role of the health-care professionals. This agent adds fall-events to the population under care based on the current fall-risk of each patient and changes the values of the patient record as it would be during the use of the EPR system. The simulation of the aging process of each patient is based on the research results of prior studies with risk-equivalent patients [3,4].

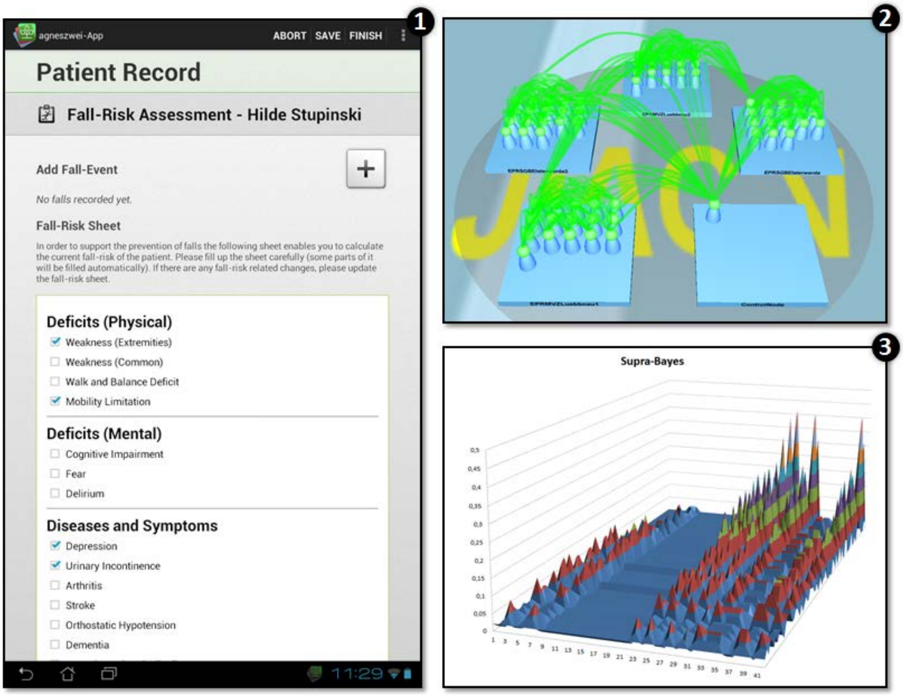


Fig. 1. (1) Screenshot of the fall-risk assessment tool as part of the EPR. (2) Visualization of the simulation environment consisting of four Tablets with (with multiple patient agents, approximately 70) and a single agent platform simulating the health-care professionals. (3) Temporal progress of the probability distribution of 50 features during the first 80 voting rounds.

In addition to the simulation we want to introduce the final fall-risk assessment tool itself as part of the agnes^{zwei} application. Fig 1.1 shows a part of the initial assessment tool enabling the health-care professional to add new fall-events and to calculate the current fall-risk of the patient. Here, we will show the notification stage of the approach reached if an elaborated feature is forwarded to the health-care professional which has to confirm whether it is a fall-risk indicator and should be added to the fall-risk assessment sheet or not.

4 Conclusion

We presented a self-learning agent-based fall-risk assessment tool as part of an EPR. The goal of the approach was to adapt the fall-risk assessment tool to the population under care to provide a more personalized health-care. We utilized an agent-based approach to model the system whereby each patient is represented by its patient agent. Consequently, the patient agents negotiate about possible fall-risk indicators. In order to demonstrate the approach we will utilize the simulation environment developed within the evaluation and show the system at runtime using the ASGARD agent viewer. Further we will present the final assessment tool as part of the agnes^{zwei} App for Tablets.

References

1. Ahrndt, S., Fähndrich, J., Albayrak, S.: Preventing elderly from falls: The agent perspective in ePRs. In: Demazeau, Y., Ishida, T., Corchado, J.M., Bajo, J. (eds.) PAAMS 2013. LNCS, vol. 7879, pp. 1–12. Springer, Heidelberg (2013)
2. Ahrndt, S., Rieger, A., Albayrak, S.: Entwicklung einer mobilen elektronischen Patientenakte für die ambulante Versorgung in ländlichen Regionen. In: Goltz, U., et al. (eds.) INFORMATIK 2012. LNI, vol. 208, pp. 1167–1181. GI (2012)
3. Fiss, T., Dreier, A., Meinke, C., van den Berg, N., Ritter, C.A., Hoffmann, W.: Frequency of inappropriate drugs in primary care: Analysis of a sample of immobile patients who received periodic home visits. Age and Ageing 40(1), 66–73 (2010)
4. Fiss, T., Ritter, C.A., Alte, D., van den Berg, N., Hoffmann, W.: Detection of drug related problems in an interdisciplinary health care model for rural areas in germany. Pham. World Sci. 32(5), 566–574 (2010)
5. Genest, C., McConway, K.J., Schervish, M.J.: Characterization of externally bayesian pooling operators. The Annals of Statistics 14(2), 487–501 (1986)
6. Goth, G.: Analyzing medical data. Communications of the ACM 55(6), 13–15 (2012)
7. Halpern, J.: From statistical knowledge bases to degrees of belief: An overview. In: Proceedings of PODS 2006, pp. 110–113. ACM, New York (2006)
8. König, H.H.: Gesundheitsökonomische Aspekte der Sturz- und Frakturprävention (March 2012), `http://www.sturzpraevention2012.de/programm_abstract.php?No=9` (last visit: February 19, 2013)
9. Lindley, D.: Theory and practice of bayesian statistics. The Statistician 32(1), 1–11 (1983)
10. Roback, P.J., Givens, G.H.: Supra-bayesian pooling of priors linked by a deterministic simulation model. Communications in Statistics – Simulation and Computation 30, 447–476 (2001)
11. Schenk, A., Ahrndt, S., Albayrak, S.: Predicting fall risks in electronic patient records. In: Goltz, U., et al. (eds.) INFORMATIK 2012. LNI, vol. 208, pp. 1194–1198. GI (2012)
12. Tonn, J., Kaiser, S.: ASGARD – A graphical monitoring tool for distributed agent infrastructures. In: Demazeau, Y., Dignum, F., Corchado, J.M., Pérez, J.B. (eds.) Advances in PAAMS. AISC, vol. 70, pp. 163–173. Springer, Heidelberg (2010)

Pedestrians and Crowd Simulations with MAKKSim - A Demonstration

Luca Crociani, Lorenza Manenti, and Giuseppe Vizzari

CSAI - Complex Systems & Artificial Intelligence Research Center
University of Milano-Bicocca
Viale Sarca 336, 20126, Milano, Italy
{luca.crociani,manenti,vizzari}@disco.unimib.it

Abstract. This paper presents MAKKSim a system supporting the modelling and simulation of pedestrians and groups in an environment. The paper shows how the MAKKSim system can be employed to generate what-if scenarios showing the plausible overall system behaviour of the simulated scenario subject to a specific crowding situation.

Keywords: Multi-Agent Systems, Complex Systems, Crowd simulation.

1 Introduction

Computer models for the simulation of crowds are growingly investigated in the academic context and employed by firms and decision makers[1]. Models and simulators have proved their adequacy in supporting architectural designers and urban planners in their decisions by creating the possibility to envision the behaviour of crowds of pedestrians in specific (already existing environments and planned) designs, to elaborate what-if scenarios and evaluate their decisions with reference to specific metrics and criteria.

2 MAKKSim Overview and Workflow Introduction

The MAKKSim model (see [4] for a thorough description) and simulation system is aimed at supporting decision makers in the evaluation of designs and crowd management solutions by means of statistical data that are the outcomes of simulations taking place in the analysed environment in specific crowding conditions. The typical situation and workflow of the MAKKSim system is shown in Figure 1: starting from CAD files, information about the observed/expected demand and additional information about pedestrians' preferences (e.g. percentage of pedestrians employing stairs, escalators or elevators in a multi–level

[1] See `http://www.evacmod.net/?q=node/5` for a significant although not necessarily comprehensive list of simulation platforms. The list includes over 60 models, commercial and academic, general purpose or specifically targeted on certain situations and scenarios, maintained or discontinued.

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 267–270, 2013.

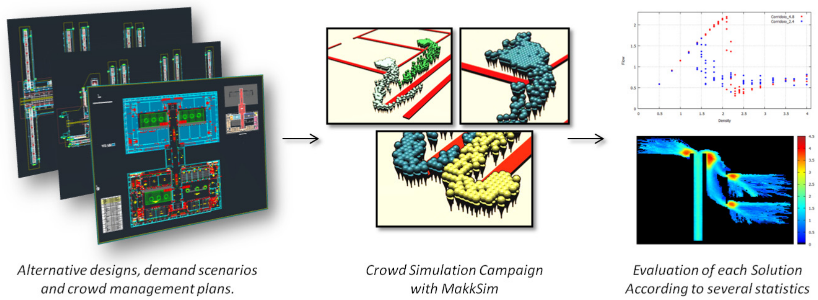


Fig. 1. The application scenario of MAKKSim: evaluation of designs and crowd management solutions by means of statistical data from simulations

scenario), the system allows the execution of simulation campaigns producing data about the plausible performance of the environment when subject to the analysed crowding conditions. MAKKSim essentially consists of a set of scripts written in the Python programming language extending the Blender 3D modelling toolkit. The workflow will be briefly introduced in the following.

2.1 Design of the Environment

In this activity the user prepares the simulation environment (i.e. the walk-able area), by defining its size and annotating it with spatial markers. By employing Blender standard functions, MAKKSim allows the import of CAD files (.dxf or .3ds format), that can be interpreted as the set of *obstacles* defining the walkable area for pedestrians by using a semi-automatic procedure (Fig. 2(a)-(b)). A cleaning phase of the imported data[2] is required; the construction and extrusion along the z–axis of the objects faces follows, starting from the edges of the cleaned model, with the aim to make them readable from the discretisation procedure and to directly transform them into spatial markers of *obstacle* type (characterised by the red colour).

The 2-dimensional environment defined in the CAD file is therefore converted into an empty environment that can be annotated by means of additional spatial markers: the latter can be simply introduced by adding new objects whose colour must be green for the start areas or blue for end points (Fig. 2(c)). These objects can be further characterised by means of properties that are however related to the specific scenario and that will be described in the following.

2.2 Configuration of the Scenario

When the environment design phase is completed, the configuration of the simulation can be done by using the MAKKSim menu (Fig. 2(d)) and the Blender

[2] For instance, double vertices and edges have to be removed and shape of obstacles must be closed.

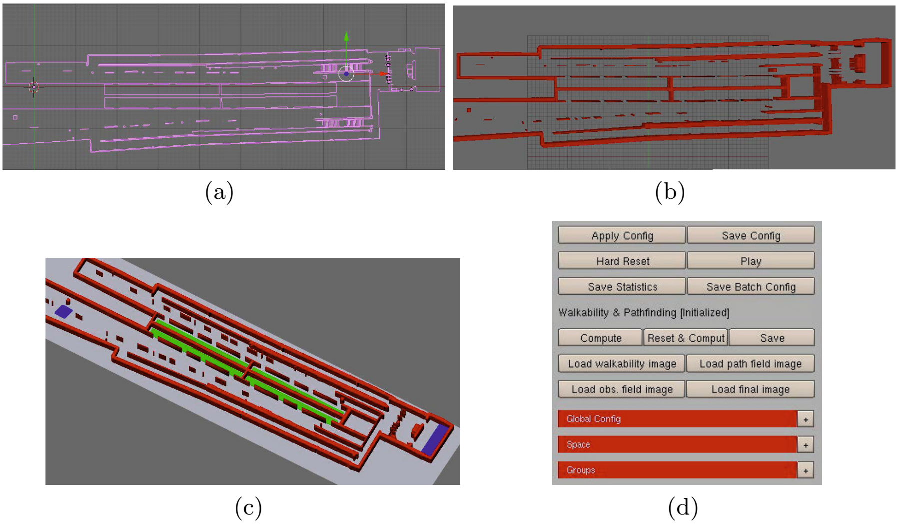


Fig. 2. Environment modelling and scenario configuration in MAKKSim: from the 2D CAD (a) the obstacles are derived (b); additional spatial markers are added (new blue or green shapes) (c); the menu for the configuration of the simulation scenario (d)

objects properties, which are used for the configuration of each *start area* and (optionally for other modelling elements. By using the menu, it is possible to set the *global configuration*, such as the type of agent representing the pedestrians, the calibration weights of the utility function of the agent, as well as duration of the simulation and interval for statistics gathering. This menu also allows setting the size of the portion of environment considered for the simulation (the grey plane in Fig 2(c) representing a boundary area for various morel algorithms) and defining the types of groups present in the simulation (size and, optionally, structure). By introducing new parameters in the start area objects, it is possible to set information such as: number of pedestrians and groups generated in it, their frequency of generation and their targets (i.e. target areas).

2.3 Simulation Execution and Statistics Evaluation

Once the scenario has been configured, the simulation can be started by pressing buttons "Compute", which executes the environment discretisation procedure and initialises the *floor fields*[3], "Apply Config" and "Run". The first simulation step is only used for the initialisation of the statistics files. The simulation actually starts at the second step, where the pedestrians generation and update begins, according to the user-defined policies.

The simulation evolution is graphically rendered with Blender, in order to have a direct feedback of the emerging situations and to make a preliminary

[3] One *path grid* for each end point, one *obstacle grid* and one *density grid*, that is updated at each step.

analysis of the modelled *what-if* scenario. Of course, the *face validation* [3] is not sufficient to evaluate the model outcomes, therefore the second phase of the scenario evaluation consists in the analysis of the statistical data. Once the simulation is finished, the latter can be saved by pressing the related button, which will generate several files, schematically related to *global* statistics (aggregation of all data), *meso level data* (e.g. pedestrians from a specific start and to an end point, only members of a group) and optionally also individual data. In fact, statistical data obtained with the simulations are based on fine-grained statistics about individual behaviours, that are the starting point for more aggregated measurements such as trajectories, travel times, pedestrian flows in sections like corridors, corners and so on. Some of the aggregated statistics are referred to the environment in which the simulation takes place: in particular, *cumulative mean densities* are important to define the so-called *level of service* [2], extremely important to understand and evaluate the performance of the design in specific crowding conditions.

3 Conclusions

This paper has briefly introduced the MAKKSim simulation system and the workflow that leads from the import of a CAD file representing an environment in which the simulation must take place, to the configuration of the pedestrian dynamics to be generated, to the execution of the simulation and storage of statistics about the outcomes. Another paper in the present volume [1] presents a more thorough description of the MAKKSim system architecture and a real world case study in which it was applied.

References

1. Crociani, L., Manenti, L., Vizzari, G.: MAKKSim: MAS-based crowd simulations for designer's decision support. In: Demazeau, Y., Ishida, T., Corchado, J.M., Bajo, J. (eds.) PAAMS 2013. LNCS, vol. 7879, pp. 25–36. Springer, Heidelberg (2013)
2. Fruin, J.J.: Pedestrian planning and design. Metropolitan Association of Urban Designers and Environmental Planners (1971)
3. Klügl, F.: A validation methodology for agent-based simulations. In: Menezes, R., Viroli, M. (eds.) Symposium on Applied Computing, pp. 39–43. ACM Press (2008)
4. Vizzari, G., Manenti, L., Crociani, L.: Adaptive pedestrian behaviour for the preservation of group cohesion. Complex Adaptive Systems Modeling 1(1) (in press, 2013)

GAMA: A Spatially Explicit, Multi-level, Agent-Based Modeling and Simulation Platform

Alexis Drogoul[1], Edouard Amouroux, Philippe Caillou, Benoit Gaudou, Arnaud Grignard, Nicolas Marilleau, Patrick Taillandier, Maroussia Vavasseur, Duc-An Vo, and Jean-Daniel Zucker

IRD, UMI UMMISCO 209, 32 av. H. Varagnat, 93143 Bondy, France
alexis.drogoul@gmail.com

Abstract. Agent-based modeling is now widely used to investigate complex systems but still lacks integrated and generic tools to support the representation of features usually associated with real complex systems, namely rich, dynamic and realistic environments or multiple levels of agency. The GAMA platform has been developed to address such issues and allow modelers, thanks to the use of a high-level modeling language, to build, couple and reuse complex models combining various agent architectures, environment representations and levels of abstraction.

Keywords: Simulation platform, Agent-based modeling, GIS data, Multi-level model.

1 Introduction

Agent-based modeling (ABM) has brought a new way to study complex systems by allowing to represent multiple heterogeneous entities interacting in a non-linear fashion in a shared environment. Although it is now used in different domains, ABM still struggles with two issues. First the lack of a comprehensive and common representation of the environment(s) in which agents interact, which limits its usefulness for models where the environment itself is to be represented as a complex entity. Secondly a difficulty to go beyond the classical Object-Oriented Paradigm to express interactions between agents at different levels of abstractions (e.g. agents composed of agents). Even though tools have been proposed in the recent years, they are too complex for domain experts to build their models without a strong support from computer scientists. For instance, building a realistic model that relies on GIS data at different geographical scales still involves complicated coding tasks in most ABM environments.

The GAMA (GIS & Agent-based Modeling Architecture) modeling and simulation platform has been proposed to address such shortcomings. On one hand, this open source platform allows the definition of agent-based models with complex environment representations and generic multi-level capabilities. On the other hand, it provides field experts, modelers, and computer scientists with a complete modeling and simulation environment for building spatially explicit agent-based models with ready-to-use abstractions for the most common needs

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 271–274, 2013.

(e.g. decision architectures, generic behaviors, such as movements regardless on the environment representation), which are accessible through a dedicated high-level modeling language (GAML) and remains extensible by Java programmers.

2 Main Purpose

GAMA is based on: (i) a meta-model [5] dedicated to complex environment representation and multi-level models; (ii) a modeling language (GAML) and its related elements (parser and compiler); (iii) an efficient virtual machine to execute model and to interact with the simulation. Compared to other frameworks such as NetLogo [6] or Repast Symphony [2], its main advantage is to provide this multi-level architecture (extendable via plug-in), and a very complex environment representation easily defined via GAML.

Multi-level Modeling. Multi-level agent-based modeling requires to manipulate agents at different levels of representation w.r.t. to time, space and behavior. Our approach of multi-level modeling is based on three principles. (1) An agent represents a level of organization which is associated with a spatial and temporal scale. (2) Levels are hierarchically organised to define privileged interactions between embedded levels. (3) Organizations can be dynamic: some agents can move from an organization to another in order to adapt their representation level dynamically.

The GAMA meta-model [5] has been designed to fulfill these three features. An agent is an instance of a species (the kind of agent, as in classes and instances in OOP). Every agent possesses a spatial representation (its shape) that is located in an environment. An agent, as organization level, can host populations of micro-agents. Its topology defines the spatial environment of hosted agents. Similarly, the macro-agent defines the execution time scale of hosted agents by specifying the way they will be scheduled. This thus defines a hierarchical structure of the model: any agent is hosted by another agent and possibly hosts populations of other agents. The macro-agent will manage the relations between its behavior and micro-agent behavior: it is indeed allowed to redefine the behavior of hosted agents. A top-level agent (the world) has been introduced. It manages the global variables and parameters of the simulation and its shape defines the reference environment. Finally, agents can dynamically migrate from a population to another one (by changing species); this allows agents to change their organization level, and thus representation level during the simulation course.

Environment Representation. GAMA is particularly powerful concerning the management of complex environments. It allows to define several environments with different topologies (grid, graph or continuous). One continuous environment is used as reference to synchronize all of them. Each GAMA agent has a shape, that is a 3D simple (point, polyline or polygon) or complex (composed of several geometries) geometry.

A particularly interesting feature of GAMA is the possibility to create agents and to define their attributes (in particular their shape) from real data using shapefiles. Conversely, this allows the modeler to integrate geographical data into models under the form of active agents (one agents is created by geometry of a shapefile). In addition, GAMA manages the spatial projection of the data (to get a spatially coherent model) and the reading of attribute values. In order to ease geometries use and manipulation, high-level geometry transformations (e.g. buffer, convex-hull, etc.) and movement primitives (e.g. shortest path computation) are readily available in GAML.

3 Demonstration

GAMA can be used for lots of purposes including teaching, conceptual modeling and applied research. We illustrate its power with applied and abstract uses[1].

Applied Models. GAMA has already been used in various large scale applications that share a strong focus on the interactions between agents and complex environments. Epidemiological models have been developed to study avian flu persistence in North Vietnam and rift valley fever propagation in Senegal. It has been used to assess the effectiveness of control policies on the recurrent invasions of insects in the Mekong delta, to simulate rescue management in Hanoi after an earthquake or evacuation organisation in case of a tsunami in Nha Trang (Vietnam). The MAELIA project [3] uses it to study socio-ecological impacts of water management policy in the Adour-Garonne Basin (France).

GAMA can manage a large number of agents for real-scale applications, for example, nearby 200 000 for the MIRO model (Figure 1) [1]. This model addresses the issue of sustainable cities by focusing on one of its very central components, daily mobility. Therefore, improving urban accessibility merely results in increasing the traffic and its negative externalities (congestion, accidents, pollution, noise...), while reducing at the end the accessibility of people to the city. For that, an ABM has been developed and applied to Dijon and Grenoble, two mid-sized cities (nearby 120 000 inhabitant) in France. The simulator is used to realise scenarios determined by geographers for quantifying, for example, service accessibility and to organise serious game sessions for identifying cities management strategies.

Abstract Models. As presented in previous section, GAMA offers several modeling capabilities, like multi-level simulation, seamless integration of GIS Data or extensible architecture with plugin. Very simple conceptual models can be developed and used for demonstration or conceptual proof. For example, the multi-level architecture can be illustrated by a flocking example with Flock agent dynamically created when nearby boids converge (an illustration can be watched in the video). The Flock agent captures boids agents and computes its own geometry from their individual data. Boids will be released when the Flock will disappear (when it moves toward an obstacle).

[1] Link to a video:
`http://code.google.com/p/gama-platform/wiki/VideosPresentation`

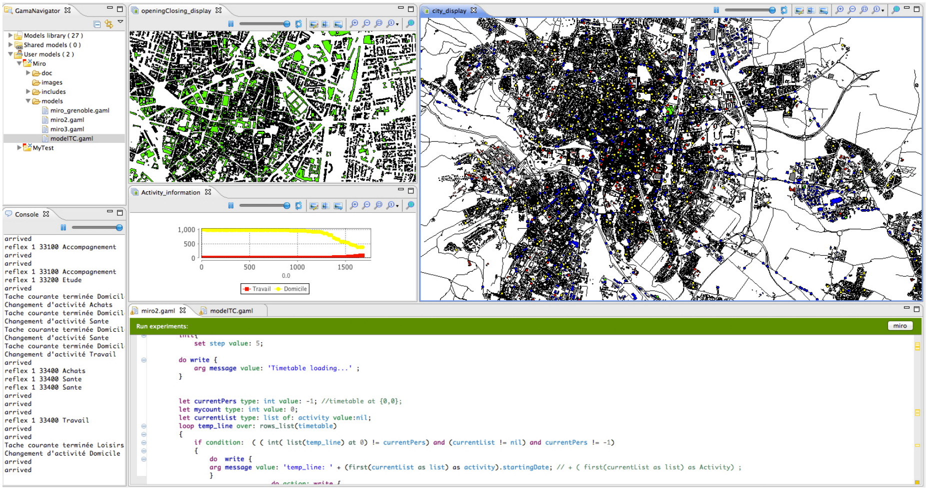


Fig. 1. MIRO simulation interface

4 Conclusion

The latest version of GAMA, bundled with a set of example models, as well as its source code, can be downloaded from its website[2]. The site also provides users with a complete modeler guide, several tutorials, the GAML language reference, as well as a guide for developers to extend the platform with their own plugins.

References

1. Banos, A., Marilleau, N., MIRO team: Improving individual accessibility to the city: an agent-based modelling approach. In: Proc. of ECCS, Bruxelles (2012)
2. North, M., Howe, T., Collier, N., Vos, J.: A declarative model assembly infrastructure for verification and validation. In: Proc. of Advancing Social Simulation: 1st World Congress (2007)
3. Taillandier, P., Therond, O., Gaudou, B.: A new BDI agent architecture based on the belief theory. Application to the modelling of cropping plan decision-making. In: Proc. of IEMs, Germany, pp. 107–116 (2012)
4. Taillandier, P., Vo, D.-A., Amouroux, E., Drogoul, A.: GAMA: A simulation platform that integrates geographical information data, agent-based modeling and multi-scale control. In: Desai, N., Liu, A., Winikoff, M. (eds.) PRIMA 2010. LNCS, vol. 7057, pp. 242–258. Springer, Heidelberg (2012)
5. Vo, D.-A.: An operational architecture to handle multiple levels of representation in agent-based models. PhD thesis. Université P&M Curie, Paris (2012)
6. Wilensky, U.: Netlogo. Technical report, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL (1999)

[2] http://gama-platform.googlecode.com

Demonstrating SERENA: Chance Encounters in the Space of Ideas

Jamie Forth[1], Athanasios Giannimaras[1], Geraint A. Wiggins[1],
Robert Stewart[2], Diana Bental[2], Ruth Aylett[2],
Deborah Maxwell[3], Hadi Mehrpouya[3], Jamie Shek[3], and Mel Woods[3]

[1] School of Electronic Engineering and Computer Science
Queen Mary University of London, Mile End Road, London, England, E1 4NS
[2] School of Mathematical and Computer Sciences
Heriot-Watt University, Edinburgh, Scotland, EH14 4AS
[3] Duncan of Jordanstone College of Art and Design
University of Dundee, Perth Road, Dundee, Scotland, DD1 4HT

1 Introduction

We demonstrate SERENA, a multi-site, pervasive, agent environment that suppers serendipitous discovery in research. SERENA attempts to assist researchers by presenting them with information that they did not know they needed to know about their research. In this demonstration, we show:

- Public SERENA: Streamed information from a library, displayed to stimulate ideas;
- SERENA user modelling: On-line discovery of user information and inspectable user model;
- SERENA suggesting: User's notes and suggestions for reading;
- SERENA intervening: Informed academic match-making by the agent system.

If possible, we will demonstrate further progress in the project beyond the above at the conference. If the networking facilities at the conference allow, delegates will be able to take part in the demos via their browsers. It will not be possible to download the SERENA Notebook App, because of technical restrictions applied by the Android App Store.

2 Public SERENA

Our first demo is of public SERENA. It is conceived in the context of a major UK city library. It is architecturally melded with the building, in that its outputs are projected directly on to walls, using site-specific designs that integrate with the architecture. Its outputs consist of simple visualisations (the simplest being mere text) of documents that are ordered via the library's on-line order system. The information is filtered, so that no connection with the library user ordering it can be made: their name is not displayed, nor is the time at which they placed their order. Various display methods are implemented. The aim of the installation

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 275–278, 2013.

(and the demo) is to stimulate ideas in people viewing, that are connected with recent activity in the Library. A simple example design is given by Forth et al. [1, Fig. 2]; here we demonstrate more abitious designs.

This demo is simple as far as the agent system is concerned. A shadow agent [1, §2.1] receives text input in an agreed format from an authenticated data source. The agent preprocesses the data into the internal RDF representation, and sends it directly to another shadow agent, which shadows the devices that are displaying the information. Authentication is available, but in this demo we omit it[1] so that conference delegates may connect and watch the demo. See `www.serena.ac.uk/paamsdemo/` for technical details of downloads to use the service.

3 SerenA User Modelling

SerenA builds a model of its user, including information given by the user, information inferred directly under the control of the user, and information inferred about the user by the system. The modelling process is kick-started by our Discover.Me.Semantically service[2], which searches for information about a new user, and then consults with them to select what is relevant. Discover.Me.Semantically is a web-based tool that allows its user to author RDF representing their professional and personal interests, skills and expertise. The stand-alone implementation allows the user to download this RDF representation as a file to be hosted on their own web pages; here, in the version avaialable from `www.serena.ac.uk/paamsdemo/`, we download your information[3] into the SerenA user model, and allow you to inspect it, using a third-party web resource designed for that purpose.

This part of the demo does not engage the agent system, but the RDF user model is crucial to SerenA's operation, so it is important underpinning technology.

4 SerenA Suggesting

A key issue in supplying the user with useful information is to understand the research goals that they are expressing in their notes, files and email[4]. We use the GATE natural language processing system[5]to detect goals in natural language sentences [2], and an ontology for goals has been defined [4].

---

[1] No sensitive information is available in the demo system.

[2] Source code under GPLv3 license:
`https://github.com/robstewart57/discover-me-semantically`
Running instance:
`http://serena.macs.hw.ac.uk/serena/discover-me-semantically/`

[3] By proceeding with the Discover.Me.Semantically process, you agree for us to store your data temporarily. However, we promise to delete it after the conference, and not to use it for anything other than our demo meanwhile.

[4] Of course, SerenA does not access files or email without permission.

[5] `http://gate.ac.uk`

This part of the demo focuses on the SerenA Semantic Notebook App [1,3] for Android. The researcher-user is invited to make notes, add tagged images, keywords, and so on, all in free text. Interaction with the user is then managed by SerenA processing the user's text, and then adding annotations (e.g., items of text, web links), to the notes, making the distinction between the user's notes and SerenA's additions clear by means of typography. Example views are given by Forth et al. [1, Fig. 1].

From the agent perspective, when the user adds a new note, it is passed by the Android device via its shadow agent into the agent user model (which is where the user's notes are stored). The agent also broadcasts the arrival of the note, as a piece of text associated with a given user, to the entire agent system. At this point, any agent that is capable of processing text may process it and attempt to do something with the result. Here, we focus on the goal-detection agent, which processes the text and, if it finds a research goal, attempts to express in RDF what that goal is. If it is successful, the goal is added to the user model (with provenance, so that it can be undone if later proven incorrect, and also so that its effect can be time-appropriate). The goal is also broadcast, identified as such in RDF, so that agents with expertise on goals may operate on it. Here we focus on a simple lookup agent, that receives the broadcast goal, recognises the action in the goal (for example, "find out about", or "look up") as a research goal, and looks up its object (the thing sought) on DBpedia. If it is successful, then it reports the fact to the user model, which takes the resulting suggestion and adds it to the notebook in the user model. As a result of doing this, a visual representation of the suggestion appears in the notebook, associated with the note that gave rise to it.

In more realistic situations than this simple demo, this kind of decoupled interaction is advantageous for two reasons: first, SerenA must avoid the paperclip effect[6]; and, second, the reasoning required to suppress pointless or dull information is often extensive, and cannot be performed on the fly, while the user waits. The concomitant advantage of asynchrony is that network outages do not degrade the experience: the user will come to expect SerenA suggestions at some time after they make their notes, but not immediately.

5 SerenA Intervening

The final part of our planned demo builds on the situation above, introducing new users, who are co-located at a conference in Salamanca. As the first user makes a note, and it content resonates around the agent community, as above, the second users' agents receive the broadcast, noting that that user has made notes on a closely related topic. Both users are informed, via their notebooks, that the other is present at the conference, and an appropriate web reference is given, to enable them to look each other up.

[6] The irritation produced when Microsoft's Clippy character used to intrude unexpectedly, distracting the user from their task, with often incorrect information.

Acknowledgements. This work is supported by EPSRC Sandpit Research Grant EP/H042741/1 – SerenA: Chance encounters in the Space of Ideas.

References

1. Forth, J., Giannimaras, T., Wiggins, G.A., Stewart, R., Bental, D., Aylett, R., Maxwell, D., Mehrpouya, H., Shek, J., Woods, M.: SERENA: A multi-site pervasive agent environment that supports serendipitous discovery in research. In: Demazeau, Y., Ishida, T., Corchado, J.M., Bajo, J. (eds.) PAAMS 2013. LNCS, vol. 7879, pp. 85–96. Springer, Heidelberg (2013)
2. Giannimaras, A.: Goal detection in natural language text. Msc thesis, School of Electronic Engineering and Computer Science, Queen Mary University of London (2012)
3. Maxwell, D., Woods, M., Makri, S., Bental, D., Kefalidou, G., Sharples, S.: Designing a semantic sketchbook to create opportunities for serendipity. In: Proceedings of BCS HCI Conference (2012)
4. Piao, S., Bental, D., Whittle, J., Aylett, R., Makri, S., Sun, X.: A pilot study: Deriving a users' goal framework from a corpus of interviews and diaries. In: Proceedings of the Workshop on Semantic Relations II - Enhancing Resources and Applications (2012)

Automatic Deployment of a Consensus Networks MAS*

Yolanda Gómez, Alberto Palomares, Carlos Carrascosa, and Miguel Rebollo

Universitat Politècnica de València,
Camino de Vera s/n 46022 Valencia Spain
{ygomez,apalomares,carrasco,mrebollo}@dsic.upv.es

1 Introduction

In agent-based networks, 'consensus' is referred to reach an agreement about a certain quantity of interest or distribution function that depends on the state of all agents [2]. Consensus algorithms can be modeled as iterative processes in which autonomous agents work in a distributed fashion, without the necessity of sending information to a central node that acts as coordinator.

Olfati-Saber and Murray [2] have demonstrated that a convergent and distributed consensus algorithm in discrete–time can be written as follows:

$$x_i(k+1) = x_i(k) + \varepsilon \sum_{j \in N_i} a_{ij}(x_j(k) - x_i(k)), \tag{1}$$

where N_i denotes the set formed by all nodes connected to the node i (neighbors of i).

The algorithm converges to the average of the initial values of the state of each agent and allows computing the average for very large networks via local communication with their neighbors on a graph. The specification of the consensus process in matricial form is:

1: $D = \sum_{i \neq j} a_{ij}$
2: $L = D - A$
3: assign a $\varepsilon < 1/\Delta$
4: $P = I - \varepsilon L$
5: init x with random values
6: **repeat**
7: $\quad x = P * x$
8: **until** the system converges or maxiter reached

The use of a matricial system assumes a complete knowledge scenario in which all agents know the entire network structure and the values. When it is implemented in a real system, some considerations about how the communication process will take place must be taken into account. The set of rules that specifies the information exchange between an agent and all of its neighbors on the network are specified in a consensus protocol.

* This work is supported by TIN2012-36586-C03-01 and PROMETEO/2008/051 projects of the Spanish government, CONSOLIDER-INGENIO 2010 under grant CSD2007-00022, and PAID-06-11-2084.

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 279–282, 2013.

2 Main Purpose

The main goal of the work here presented is the easy deployment in real multi-agent systems of consensus processes. MAGENTIX2 agent platform [4] has been used to implement real agents that follow the consensus algorithm to reach agreements [3].

The topology of the multi-agent connection network is defined by the system developer, and a GraphML file with such topology is generated. This file is the input to the mechanism that generates automatically the MAS following such connections (implemented in MAGENTIX2) and execute it to reach a consensus (see Figure 1).

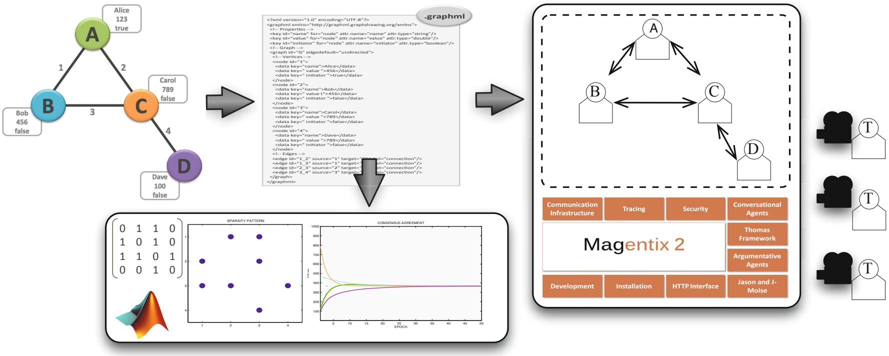


Fig. 1. Implementation process developed

Agents that participate in the consensus process make use of the MAGENTIX2 tracing services [1] to generate events with all the needed information about the consensus process. So, any number of monitoring agents can be generated to connect to the events generated by such agents. These monitoring agents have their own Ubigraph server, a free software for visualizing dynamic networks. The server is provided with all the information from the events generated by the agents involved in the consensus process. In this way, it is easy to follow the consensus process and check if and how the MAS reach a consensus through the visualization of the graph using a color codification to describe the current value of each agent and different shapes of the nodes to describe the state of each agent in the consensus protocol.

This GraphMl file is also used to realize the simulation calculus using MATLAB to test in a first installment the validity of the proposal.

3 Demonstration

In order to show the above described mechanism, a simple demo have been designed with 25 agents. These agents are structured in a random network with average degree $d = 8$.

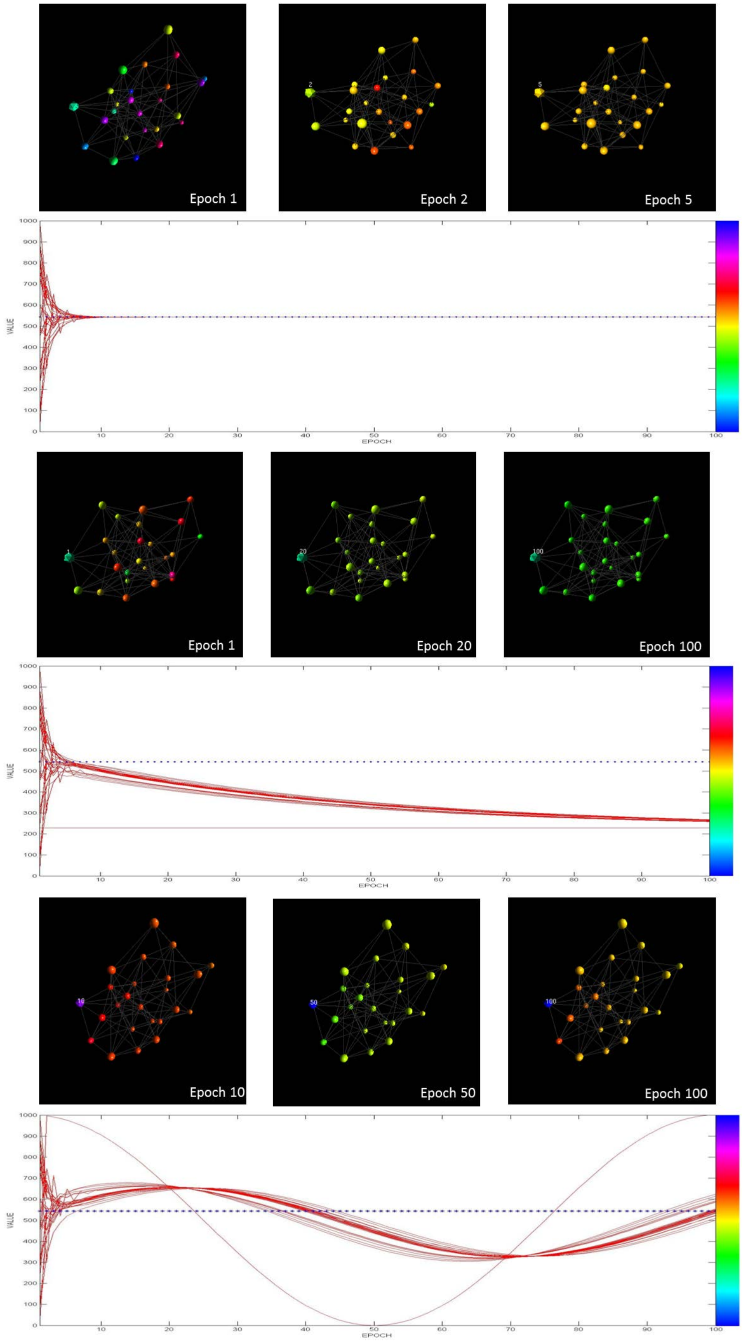


Fig. 2. Evolution of the consensus network in the MAGENTIX2 implementation

Three scenarios are shown, where $\varepsilon = 0.1$, during 100 epochs, and all the initial values of the agents are uniformelly distributed in [0, 1000]. Figure 2 shows the real execution of the same consensus process in the MAGENTIX2 implementation. Some snapshots of the animation generated in real–time by the monitoring agents have been taken in different time instants. The color of each node reflects the current consensus value, according to the color scale that appears on the right.

Scenario 1: Regular Consensus This scenario shows the normal working of a consensus network (equation 1). The average value is reached approximately in 10 epochs (Fig. 2 top).

Scenario 2: Static Regulator Agent Consensus This scenario shows how the consensus process can be used to regulate the network, just by having a static regulator agent, that have a constant value that is reached by the rest of the network (Fig. 2 middle). The presence of such an agent makes it unnecessary to monitor the values of all the agents in the network to assure that this regulation is carried out. The justification is that if a single agent leaves its value unchanged, all others will asymptotically agree with the leader according to the consensus protocol and an alignment is reached [2].

Scenario 3: Dynamic Regulator Agent Consensus This scenario is similar to the previous one, but using a regulator that changes its values dynamically. At each moment, all the agents try to reach the regulator agent reference value (Fig. 2 bottom).

4 Conclusions

This paper shows the application of consensus networks in a distributed and self-organized fashion, implemented in an agent platform with real agents.

The contribution of the paper is the proposal of a protocol that allows a network of agent achieve agreements using the consensus algorithm. This protocol has been implemented in MAGENTIX2 and tested.

These scenarios have shown the possibilities of the developed system to simulate and execute in a real MAS the same connection network to reach an agreement. Nevertheless, intensive tests have been run in the simulated process and in the MAGENTIX2.

References

1. Búrdalo, L., García-Fornes, A., Julian, V., Terrasa, A.: TRAMMAS: A tracing model for multiagent systems
2. Olfati-Saber, R., Fax, J.A., Murray, R.M.: Consensus and cooperation in networked multi-agent systems. Proceedings of the IEEE 95(1), 215–233 (2007)
3. Rebollo, M., Palomares, A., Carrascosa, C.: Towards building agreement spaces using consensus networks. In: 1st International Conference on Agreement Technologies, pp. 42–56 (2012)
4. Such, J.M., García-Fornes, A., Espinosa, A., Bellver, J.: Magentix2: A privacy-enhancing agent platform. Eng. Appl. Artif. Intel. 26(1), 96–109 (2013)

Using MASH in the Context of the Design of Embedded Multiagent System

Jean-Paul Jamont and Michel Occello

Université de Grenoble-Alpes, LCIS, 26000 Valence, France
firstname.lastname@lcis.grenoble-inp.fr
http://lcis.grenoble-inp.fr/cosy/

Abstract. MASH (MultiAgent Software/Hardware tool) assists the designer during design and deployment of the multiagent system. It enables simulation and execution of embedded multiagent systems including real world software/hardware agents according to realistic physical models.

Keywords: embedded MAS, deployment, simulation.

1 Motivation

Context. Real world embedded systems have long been small scale and standalone. Nowadays they are increasingly designed as decentralized intelligent systems. They can be considered therefore as as embedded multiagent systems (eMAS). Application domains are numerous: home automation, wireless sensor networks, collective robotics, logistics, RFID based application etc.

Problem. Designing and deploying such eMAS is a difficult task[1], because MAS development tools focus only on software. It is necessary in this context: - to test agent strategies with realistic models of physical environments because efficiency of the multiagent system is often strongly related with these models; - to be able to control the functional deviation between a virtual agent (agent simulated with a traditional multiagent simulator) and its real world embedded implementation (real world embedded agent i.e. the embedded software and its electronic board).

Contribution. MASH (MultiAgent Software/Hardware tool) assists the designer during the design and the deployment of the multiagent system. It enables simulation and execution of embedded multiagent systems :

- including real world software/hardware agents. Behavior of these agents are computed by the real world platform and injected in MASH;
- using realistic physical models at multiagent level (environment models, wave propagation models etc.) or at agent level (energy consumption model etc.)

[1] In the main track, a companion paper gives detail of these difficulties and presents the related works.

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 283–286, 2013.

2 Demonstration

The demonstration allows to gradually highlight two key features of MASH: (1) the virtual agents and real world mixed agents society and (2) the use of realistic physical models. Applications showed during the demonstration concern communication management in wireless instrumentation systems, an indoor comfort control multiagent system [1] and a prey/predator game.

Step 1: Simulation of the Virtual MAS We start by presenting MASH with a scenario involving 600 sensor agents. The agents implement the MWAC[2] model [2]. We inspect internal state of the agents (fig. 1) and the event journal. We also observe the evolution of criteria (volume of sended frame, energy consumption etc.) at an agent level and at a system level.

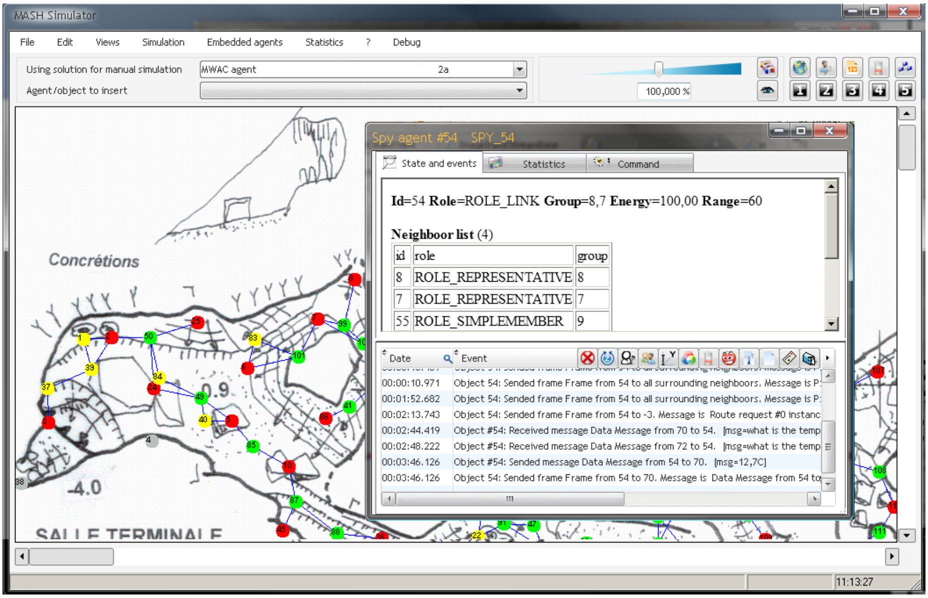


Fig. 1. Render of a multiagent systems and window of an agent internal state inspection

Step 2: Simulation of eMAS. We include the real world computed behavior of agents. In the previous set of agents, we decide to replace two virtual agents by two real world agents described in table 1. We launch the same scenario that in the first step.

We present how we use "avatars" to represent real world embedded agents in the virtual society. A video[3] of a prey/predator game (fig. 2) is presented.

[2] Multi-Wireless-Agent Communication model.

[3] This simulation require non transportable physical devices.

Table 1. The real world agents

	Real world agent 1	**Real world agent 2**
Plateform	Microchip picdem 2+	Samsung Galaxy S3
OS	No OS	Android 4.0.4 "Ice Cream Sandwich"
CPU	8-bit monocore processor PIC16F1827	32-bit multicore processor ARM Cortex-A9
CPU frequency	8MHz	1.4GHz
Memory	32kb	1Gb

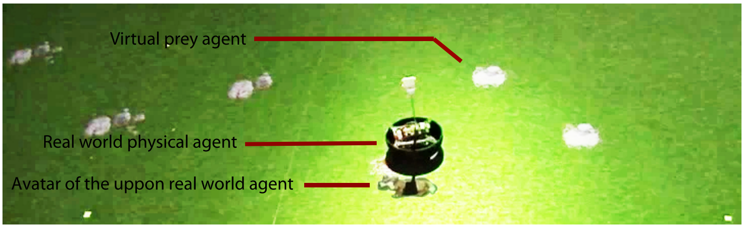


Fig. 2. Screenshoot of the projected prey/predator video

This application has been made conjointly with the CEA-LETI/Grenoble laboratory (Kurasu project). A physical robot interacts with virtual robots projected in its physical environment. The preys are virtual agents. The robot controller is build on a Virtex 4 FPGA chip (Xilinx XC4VFX20) which has a PPC405 microprocessor logical block. The position of the robot is measured with an experimental UWB location system [3]. This information is injected into MASH trough its virtual avatar. We can so see the impact of uncertainty of the location system in the simulation (the avatar should be totally covered by the real world physical agent).

Step 3: Introducing Realistic Environment Model in the eMAS. We focus on the interactions of agents situated in an indoor building. The global aim is to control the rooms temperature to ensure inhabitants comfort. We introduce a realistic physical model (detailed in [1]). This physical model, that has been used to construct a generic approach for building simulation and control development is based on the classical lumped capacity model for each considered element of the building.This approach is a network based approach as used with electrical circuits. Thermal components are used: thermal conductivity, thermal (heat) capacitance. The temperature of a room depends on one hand of elements that are only dependent on that particular room and of external parameters, and on the other hand of shared components, such as walls and doors, with other rooms.

The resulting modular model is implemented into MatLab/Simulink. Figure 3a shows the entire model of the 6 rooms building that we simulated. Figure 3b shows how we computed the temperature of the rooms and of the internal walls composing the building according to equations described in [1] that are in state space form. Figure 3c shows how the controller designed for the water temperature control depends on the different heat losses (heating floor, convector...).

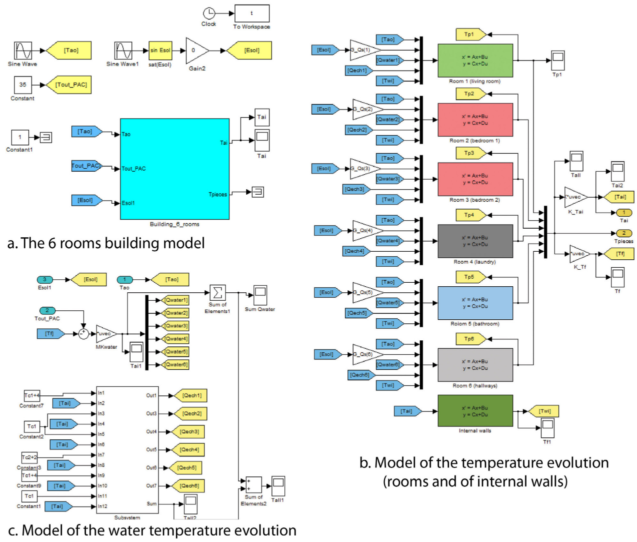


Fig. 3. Environment physical model : Part of the Matlab block model

Agents access to the environment named parameters. Perceived parameters are read in a XML files or through the Matlab client/server connection. According to agent perception abilities and agent location, MASH delivers parameters values. When agents act on the environment, the values associated to action parameters are injected into the MatLab model by the same way.

References

1. Jamont, J.P., Mendes, E., Occello, M.: A framework to simulate and support the design of distributed automation and decentralized control systems: application to control of indoor building comfort. In: Proceedings of the IEEE Symp. on Computational Intelligence in Control and Automation, pp. 80–87. IEEE (2011)
2. Jamont, J.P., Occello, M., Lagrèze, A.: A multiagent approach to manage communication in wireless instrumentation systems. Measurement 43(4), 489–503 (2010)
3. Occello, M., Jamont, J.P., Guillermin, R., Pezzin, M.: A multiagent approach for an uwb location embedded software architecture. In: 5th Int. Conf. on Soft Computing as Transdisciplinary Science and Technology, pp. 279–285. ACM (2008)

A Brownian Agent Model for Analyzing Changes in Product Space Structure in China

Bin Jiang[1,2], Chao Yang[1,2], Shuming Peng[1], Renfa Li[1], and Takao Terano[2]

[1] College of Information Science and Engineering, Hunan University, Changsha, Hunan, China
[2] Department of Computational Intelligence and Systems Science, Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, Yokohama, Kanagawa, Japan
{jiangbin,yangchao}@trn.dis.titech.ac.jp, terano@dis.titech.ac.jp

1 Introduction

The past decade witnessed an advancement of researches on complex systems by the rise and development of new network science. With respect to product space evolution and economy development issues, Hidalgo, Barabasi and Hausmann introduced an outcome-based measurement to quantify proximities among products that countries export, and used network representations to visualize proximities among products [1]. Through which, they applied theory and methods from physics and economics to model and map the impact of "The Product Space" on the development of nations [1]. The map depicts co-export patterns of 775 industrial products exported by 132 countries during 1998-2000. With the help of network expression, they found that "most upscale products are located in a densely connected core while lower income products occupy a less connected periphery" [1].

On the other hand, Schweitzer has considered wage levels as the important factor which guides the motivation of people migration and cooperative production in the economic development processes. He investigated the migration and economic agglomeration through Brownian agent approach [2]. Brownian agents are minimalistic agents with internal degrees of freedom, through specific movements, they are able to "generate a self-consistent field which in turn influences their further movement and behavior" [2].The non-linear feedback between the agents and the field generated by themselves results in an interactive structure formation process on the macroscopic level. As a result, he had found "the establishment of distinct economic centers out of a random initial distribution" [2].

Our model is essentially based on Schweitzer's work, the conceptual process and core features are similar but modified to study product space formation and economic agglomeration. Labor resources of an enterprise or a firm are simulated as Brownian agents with their internal state either employed or unemployed. They make migration decision based on their internal state and regional conditions of product space. In order to conduct the simulation model, we make two assumptions: (1) only those agents with their internal states unemployed can mobile, (2) agents always make intra-regional movements unless they move out of the boundary. Those unemployed

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 287–291, 2013.

agents make migration decision as a response to the regional product space situations. In terms of movements between two product spaces, we assume that an agent leaves a product space and arrive to another product space without time consumption.

The innovation and extension of our model are (1) it estimates the changes of product space structure by simulating labor movements inside a single product space and between two product spaces at the same time; (2) it discusses a nation's product space development from quantitative and qualitative analysis; and finally (3) it visualizes the dynamic evolution process of product space structure by Brownian agents' movements within forced-directed layout [3].

2 Purpose

The purpose of this model is to (1) simulate the product space structure formation and evolution in China, (2) investigate the product space development of China from both quantitative and qualitative analysis, and (3) visualize the dynamic growing process of Chinese product space network. We also focused on the methodology: the extended Brownian agent model, which simulates labor movements inside a single produce space and between two product spaces at the same time. This method can improve our current practices for systematically exploring both intra- and inter-regional migration under a multi-region system. The method can be broadly used in analyzing the network represented complex systems.

3 Demonstration

In our demonstration, the dynamic of product space network was visualized with forced-directed layout. According to Leamer classification [1], the overall product space classes are split into ten categories: Chemicals, Machinery, Capital intensive, Labor intensive, Cereals, Animal agriculture, Tropical agriculture, Forest products, raw materials, Petroleum. For products that cannot be classified into one of above ten categories, we classify them into a specific category. As shown in Fig.1, the above categories are denoted by nodes with different colors, where nodes represented actual products introduced by the World Trade Dataset [1]. Edges were gradually added when Brownian agents moved between nodes. The size of the node is proportional to the exports of the year.

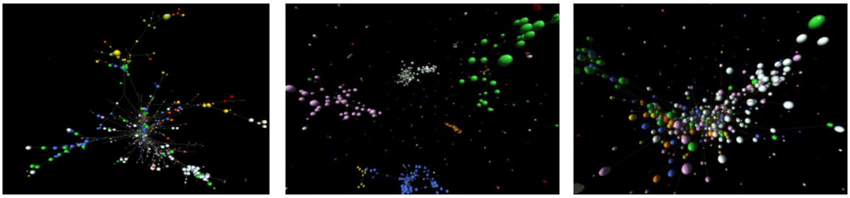


Fig. 1. Snapshots of the network structure of product space in China

Fig.1 gave the product space structure of overall categories by employing maximum spanning tree and then link the edge whith proximity threshold as 0.5 (a), only link edges in same category whith proximity threshold as 0.5 (b) and only link edges in different categories whith proximity threshold as 0.5 (c) in China, respectively. We found that the color was not randomly assigned, and similar nodes of same category are usually relatively close to form clusters of the product space network.

Fig.2 showed the result of movements and economic agglomeration insider a random selected product space, where the solid circles represent employed agents and hollow circles refer to unemployed agents. (a),(b),(c) represent distribute of brownian agents for three different time. It is obvious that agents would mobile and aggregate in some certain locations for higher economic reward.

(a)t=0 (b) t=30 (c) t=5000

Fig. 2. The agent movements inside a single region

We investigated the product space development of China from quantitative and qualitative perspective. Fig.3 gave the snapshots of evolution of overall product space in China, (a) is the inition state, it inited all products in 1997, and nodes distribute randomly. (b) is the center of the network ,with the movements of agent the center of network was formed slowly. With the further movements of agent, dissociative nodes added into the center and then the structual of network was formed(c). Fig.4 was the result of competitive product space, (a) is the inition state, it inited advantage products in 1997, and nodes distribute randomly.(b) represented some products gathered in local area. Part culsters would happened with the movements of agents then a bigger cluster could be formed (c). As time went on, the unemployed agents among the active industry would move to other regions. These mobile made some nodes start to be connected, and finally, we observed the clustering of the nodes, and non-uniform development processes between different product industries.

Fig.5 represents the change of rate of two states agent in 11 categories, θ represents the change of rate of two state agent after the simulation of the computer. It shows the rate of employed agents in labor (category 9, green node) descend, and umemployed agent rise. Machinary (category 10, white)is the opposite. Other categories remain unchanged. We can learn that although labor intensive is the leading product of China, it reaches saturation. In order to get higher reward some labor change their viewpoint(agent change its state),and then move to machinary.

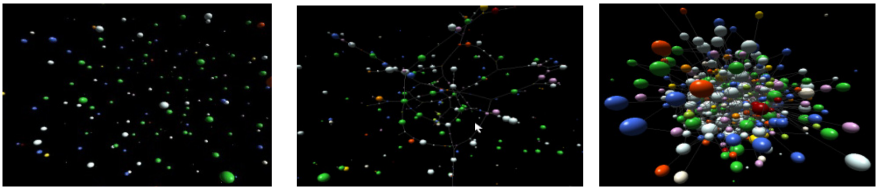


Fig. 3. Snapshots of evolution of all product space in China (quantitative analysis)

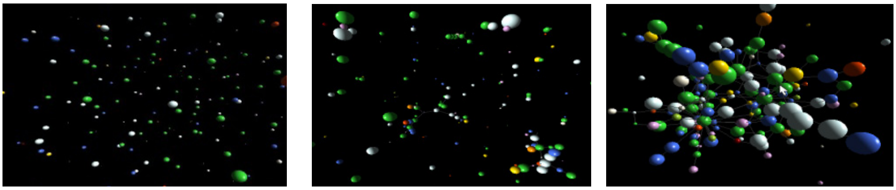


Fig. 4. Snapshots of evolution of competitive product space in China (qualitative analysis)

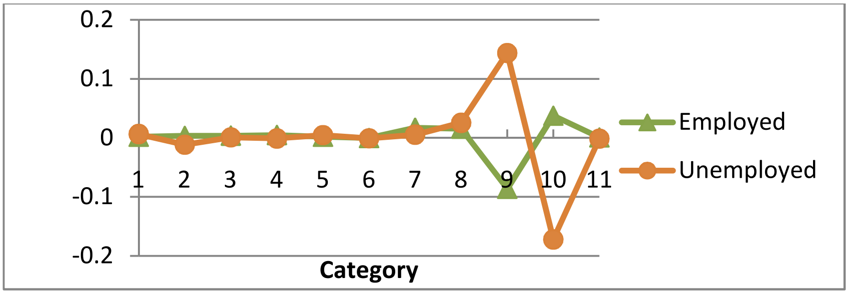


Fig. 5. The change of rate of two states agent in 11 categories

4 Conclusion

The product space structure conditions a continued structural transformation and growth of a nation's economy. While some products have more adaptable technology, labor, capital, institutions, and skills required by the new productions than others. A measurement of proximity has been employed to quantify a relationship between two products and proved to be useful in product space analysis.

This study estimates changes in product space through a Brownian agent model, by employing the proximity measurement. Labors are regarded as Brownian agents; they move through different product spaces for higher economic rewards. By means of computer simulations, the result showed that a certain amount of centralized trade areas are self-organized through Brownian agent migration and cooperative production. By employing the visualization techniques of forced-directed layout on the spatial network structure, we observed the clustering of the nodes, and non-uniform development processes between different products.

References

1. Hidalgo, C.A., Klinger, B., Barabási, A.-L., Hausmann, R.: The Product Space Conditions the Development of Nations. Science 317(5837), 482–487 (2007)
2. Schweitzer, F.: Modelling Migration and Economic Agglomeration with Active Brownian Particles. Advances in Complex Systems 1(1), 11–37 (1998)
3. Walshaw, C.: A Multilevel Algorithm for Force-Directed Graph Drawing. Journal of Graph Algorithms and Applications 7(3), 253–285 (2003)

ArgCBR-CallCentre: A Call Centre Based on CBR Argumentative Agents

Jaume Jordán, Stella Heras, Soledad Valero, and Vicente Julián

Universitat Politècnica de València. Departamento de Sistemas Informáticos y Computación, Camino de Vera s/n. 46022 Valencia, Spain
{jjordan,sheras,svalero,vinglada}@dsic.upv.es

Abstract. A good customer support service helps companies to differentiate themselves from competitors. Usually, this service is provided through a call centre that receives the users' incidences. In a call centre, there are technicians that provide customers with technical assistance. In this work, we present a call centre application deployed by using a web interface that acts as a frontend of a case-based argumentation infrastructure running on a multi-agent platform. With this application, agents representing technicians can argue to reach agreements and jointly solve incidences.

1 Introduction

Companies need to differentiate themselves from competitors by offering good customer support services. These services depends, in many cases, on the experience and skills of the technicians. A quick and accurate response to the customers problems ensures their satisfaction and a good reputation for the company.

A common customer support system consists of a network of technicians that must solve the incidences or *tickets* received in a call centre. In these centres, there are technicians whose role is to provide the customers with technical assistance. Commonly, the staff of a call centre is divided into different levels, such as: Base operators, who receive customer queries and answer those ones from which they have background training; Expert operators, who are in charge of solving complex problems; and Managers, who are in charge of organizing working groups.

To reuse previous solutions applied to each problem and the information about the problem-solving process could be a suitable way to improve the customer support offered by the company. The suitability of Case-Based Reasoning (CBR) systems in helpdesk applications to manage call centres has been guaranteed for the success of some of these systems from the 90s to nowadays [4,3]. These approaches propose systems for human-machine interaction where the CBR functionality helps the technicians to efficiently solve problems by providing potential solutions.

However, to integrate all the knowledge in a unique CBR module can be complex and costly in terms of data mining (due to large case-bases with possible

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 292–295, 2013.

out-of-date cases). Moreover, to have a unique but distributed CBR could be a solution, but to assume that all technicians are willing to share unselfishly their knowledge with other technicians is not realistic. Furthermore, several experts could provide different solutions and hence, they need a mechanism to reach an agreement about the best solution to apply. In this case, the modelling of the system as a Multi-Agent System (MAS) will be adequate.

An autonomous agent should be able to act and reason as an individual entity on the basis of its mental state (beliefs, desires, intentions, goals, etc.). As member of a MAS, an agent interacts with other agents whose goals could come into conflict with those of the agent. In addition, agents must have the ability of reaching agreements that harmonise their mental states and that solve their conflicts with other agents by taking into account their social context. Argumentation is a natural way of reaching agreements between several parties. Therefore, the argumentation techniques can be used to facilitate the agents' autonomous reasoning and to specify argumentation strategies between them [2].

2 Main Purpose

In this work, we present the ArgCBR-CallCentre tool, which provides support to the technicians of a call centre via a helpdesk application. This tool is based in the case-based argumentation infrastructure for agent societies [1] provided by the agent platform for open MAS Magentix2[1]. Thus, the argumentative agents are able to engage in argumentation dialogs to reach agreements about the best solution for a problem that must be solved, taking into account their social context. In this way, the argumentative agents can aggregate their knowledge to provide a join solution that takes the maximum profit from their problem solving experience.

3 Demonstration: Call Centre Application

The technicians of a call centre are represented by argumentative agents that argue to solve an incidence. Every agent has individual CBR resources and preferences over values (e.g. savings, quality, solving speed). A solution to a problem promotes one value, which represent the underlying reason that an agent has to prefer one solution over another one. Thus, each agent has its own preferences to choose a solution to propose. Furthermore, agents can play three different roles: *operator*, *expert* and *manager*. Also, dependency relations between roles could imply that an agent must change or violate its value preference order. For instance, an expert could impose their solutions to an operator due to a hierarchy relationship.

The application provides a web interface in which is possible to properly configure the ArgCBR-CallCentre: number and roles of agents, agent preferences, groups, etc. Moreover, the interface also allows to introduce a ticket and launch

[1] `http://www.gti-ia.upv.es/sma/tools/magentix2/index.php`

the system to obtain a solution. In addition, it is possible to know how that solution was obtained, that is, what agents were involved in the argumentation process and inspect the argumentation process itself.

In order to show how the application works, the problem-solving process to solve each ticket is shown in Figure 1 and described below:

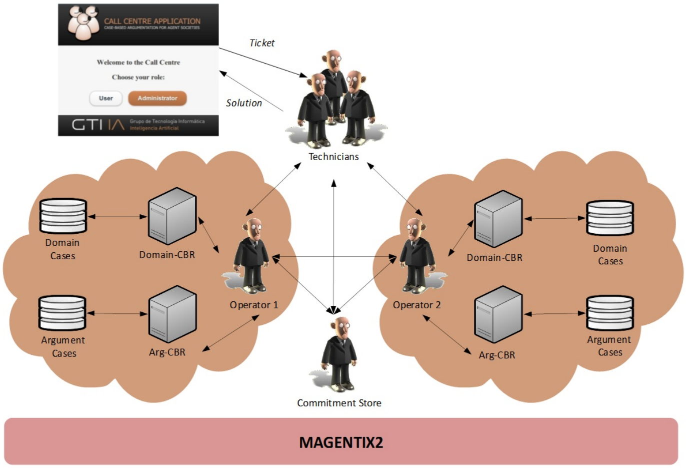


Fig. 1. Argumentation process of the call centre application

1. The process begins when a ticket that represents an incidence is registered in the web interface. Then, the argumentative agents that represent technicians are launched in the platform, taking their configuration from the web interface.
2. Each agent evaluates if it can engage in the argumentation process offering a solution. To do that, the agent makes a query to a domain CBR module to obtain potential solutions to the ticket based on solutions applied to similar tickets. If one or more valid solutions are retrieved, the agent will be able to defend a position in the argumentation dialog. Moreover, the agent makes a query to an argumentation CBR module for each possible position to defend in order to obtain its *suitability degree*. This degree represents if a position will be easy to defend based on past similar argumentation experiences. Then, all positions to defend are ordered and proposed from more to less suitability degree.
3. When agents have a position to defend (a proposed solution), these positions are stored by a Commitment Store agent. By making queries to this agent, other agents can check the positions of all dialog participants. Every agent tries to attack the positions that are different from its position. If different attack arguments can be generated, agents select the best one to rebut the position of other agents by making a query to their argumentation CBR. In this way, agents can gain knowledge about how each potential attack

arguments worked to rebut the position of an agent in a past argumentation experience with a similar social context.
4. The dialog finishes when certain time has passed without new positions or arguments proposed. If there are several positions available, the position that has been agreed by most agents is selected as the final solution. Then, this solution is communicated to the participating agents.
5. Finally, each agent updates its argumentation CBR with the new arguments produced in dialog and its domain CBR with the final solution achieved. Moreover, that solution is sent to the web interface.

This application has been tested with real data about tickets reported in a company that offers technical customer support for hardware and software problems.

4 Conclusions

In this work a customer support application based on a case-base argumentation infrastructure for MAS is presented. This application provides a web interface, capable of being executed in several devices, which allows to properly configure the system. More information and a demonstration video are available at the project web page[2]. The behaviour of the argumentative agents of our call centre has similarities with the behaviour of a human society in a real company. For example, it is possible to simulate how experts can impose their opinion over other operators, even if these experts proposed incorrect solutions. Thus, the ArgCBR-CallCentre tool can also be used to study human societies and emergent behaviours due to their social context.

Acknowledgement. This work is supported by the grants CONSOLIDER INGENIO 2010 CSD2007-00022, TIN2012-36586-C03-01, TIN2011-27652-C03-01 and PROMETEO 2008/051.

References

1. Heras, S., Jordán, J., Botti, V., Julián, V.: Argue to Agree: A Case-Based Argumentation Approach. International Journal of Approximate Reasoning 54(1), 82–108 (2013)
2. Heras, S., Jordán, J., Botti, V., Julián, V.: Case-based Strategies for Argumentation Dialogues in Agent Societies. Information Sciences 223(20), 1–30 (2013)
3. Roth-Berghofer, T.R.: Learning from homer, a case-based help-desk support system, pp. 88–97 (2004)
4. Watson, I.: Applying case-based reasoning. Techniques for enterprise systems. Morgan Kaufmann Pusblishers, Inc. (1997)

[2] `http://www.gti-ia.upv.es/sma/tools/CallCentre/index.php`

Analysis of International Relations through Spatial and Temporal Aggregation

Robin Lamarche-Perrin[1,2], Yves Demazeau[1,3], and Jean-Marc Vincent[1,4]

[1] Laboratoire d'Informatique de Grenoble, France
[2] Université de Grenoble
[3] CNRS
[4] Université Joseph Fourier
Robin.Lamarche-Perrin@imag.fr, Yves.Demazeau@imag.fr,
Jean-Marc.Vincent@imag.fr

Abstract. A leading topic of the GEOMEDIA project [1] focuses on the analysis of international relations through print media. In this demo paper, we show how techniques for Multi-Agent Systems aggregation [2,3] can be applied to the spatial and temporal aggregation of news. Two experiments show that the generated multi-resolution representations may draw attention to critical areas for the analysis.

Keywords: Interdisciplinary project, agent-based model for social sciences, news analysis, spatial and temporal aggregation.

1 Introduction

GEOMEDIA is a French interdisciplinary project of the ANR (*Agence Nationale de la Recherche*) interested in geographical and temporal analysis of news [1]. Geographers and media experts from the CIST (*Collège International des Sciences du Territoire*) collaborate with computer scientists from the LIG (*Laboratoire d'Informatique de Grenoble*) in order to design tools for the collection, the visualization and the analysis of information extracted from on-line newspapers.

In this demo paper, we focus on a major scientific issue regarding multi-scale events: how to grasp the news on different scales, from local daily events to global events that cover space and time? The aggregation technique presented in [2,3] allows to build multi-resolution descriptions of Multi-Agent Systems (MAS). Thanks to an agent-based model of international relations (section 2), we apply this technique to the aggregation of space and time (sections 3 and 4), thus revealing patterns on different scales. These patterns can be use by the domain expert to detect irregularities and events of particular interest.

2 Agent-Based Model of International Relations

A leading research topic of the GEOMEDIA project focuses on multi-scale events within international relations. In that context, we make the assumption that citations and co-citations of countries within the newspapers articles are good

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 296–299, 2013.

indicators to represent such relations. For example, the number of citations may indicate if a country is likely to politically interact with the newspaper's country (*direct links*). The number of co-citations may represent the weight of international relations *according to the newspaper's country* (*indirect links*).

Our agent-based model has two dimensions:

- The agents of the model represent 168 countries that have been selected by geographers depending on their significance for the analysis of international relations.
- The temporal dimension contains 43 weeks, from the 26th of March 2012 to the 20th of January 2013. This preliminary aggregation to the week level aims at reducing the chaotic variations of the day level.

In the following experiments, we work with the RSS feed of the *Philippine Daily Inquirer*. It has published 25,858 articles during the analyzed period. We look for the occurrences of the country names, the country adjectives, and the inhabitants names (*e.g.*, "Spain", "Spanish", and "Spaniard(s)" for the `Spain` agent). For each country c and week w, we count the number of articles that "cite c during w". 58.7% of the articles contain at least one occurence and a total of 21,541 citations have been found (1.4 citations/article in average).

In the following sections, these dimensions are aggregated with the *best-aggregation algorithm* presented in [2,3]. It builds consistent multi-resolution descriptions by aggregating redundant information within homogeneous agents while preserving the more information within heterogeneous agents. For these experiments, we used two information-theoretic measures: Kullback-Leibler divergence and Shannon entropy reduction [2]. The aggregation parameter p varies from $p = 0$ (microscopic level: neither divergence, nor entropy reduction) to $p = 1$ (full aggregation: maximal divergence and entropy reduction).

For a web-application perspective, the database has been developed with *MySQL* and the aggregation technique implemented in *PHP*. The plots and the maps of this paper have been rendered with *R*.

3 Spatial Aggregation of Countries

In this first experiment, the temporal dimension is fully aggregated. The `WUTS` hierarchy – used by geographers to build global statistics about world areas – defines groups of agents according to the world topological properties. `WUTS` is then exploited to build multi-resolution organizations that preserves the geographical properties of the agent dimension [2]. The top-left map in Fig. 1 shows the citations numbers for each of the 168 agents (microscopic level). As the aggregation parameter p increases, some agents are aggregated to give a macroscopic view of the system.

- *Homogeneous* groups of agents are aggregated in order to suppress redundant information. *E.g.*, the `Latin America` and `Sub-Saharan Africa` groups (where countries are not much cited) and the `Western Europe` group (where countries are all much cited) are aggregated for $p = 0.25$ and higher.

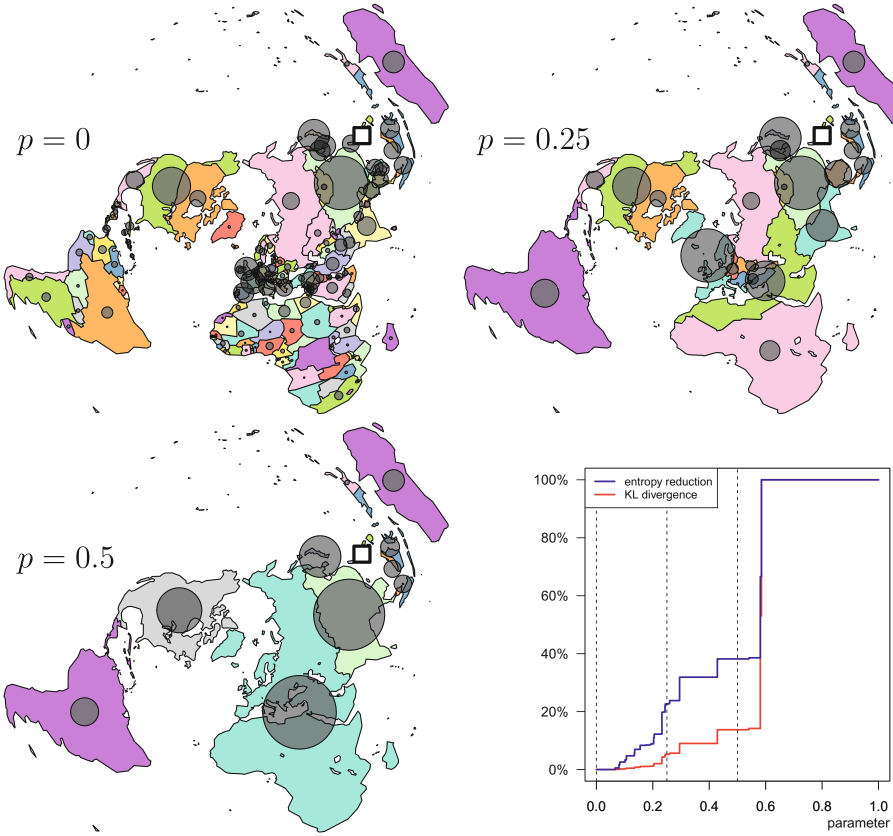


Fig. 1. Spatial aggregation of countries: the area of black circles is proportional to the number of citations of the corresponding agent, or group of agents. The plot gives the entropy reduction and the divergence of generated maps.

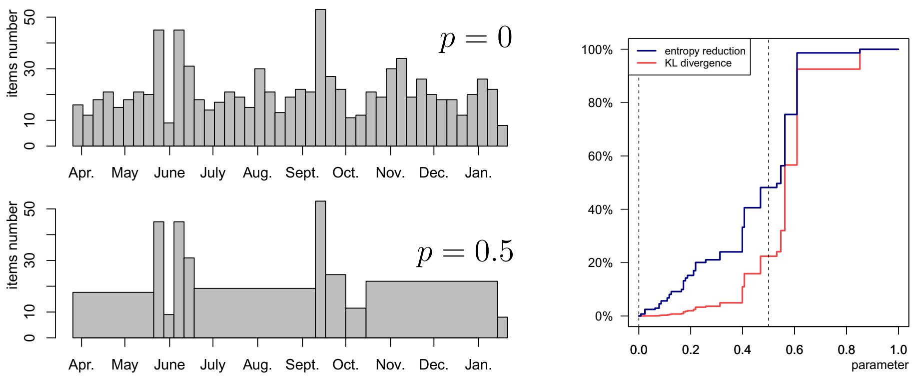


Fig. 2. Temporal aggregation of weeks: the area of rectangles is proportional to the number of citations during the corresponding period. The plot gives the entropy reduction and the divergence of generated barplots.

- On the contrary, details regarding *heterogeneous* groups are preserved by the aggregation in order to prevent information losses. *E.g.*, the `Southern Asia` group is detailed for $p = 0.5$ and lower. The huge citations number of `Japan`, `China` and `Philippines` makes indeed the aggregation not suitable. The plot of Fig. 1 show that divergence dramatically increases for $p > 0.6$ (when the `Southern Asia` group is finally aggregated).
- The `North America` group constitutes an intermediate case. Details are preserved for $p = 0.25$, so the user can see that the `USA` agent is much more cited than the others. However, it is aggregated for $p = 0.5$, but not with the `Latin America` group. This indicates that theses two high-level groups are *globally heterogeneous.*

4 Temporal Aggregation of Weeks

The top-left barplot of Fig. 2 gives the temporal variation of the citations number for the `China` agent. The bottom-left barplot gives the aggregation for $p = 0.5$. It reveals three stationary periods where `China` is more or less cited 20 times a week. It also reveals two unstable periods where `China` is very irregularly cited:

- The first peak, at the end of May, may be related to new developments in the "Scarborough Shoal standoff" between China and the Philippines.
- The second peak, in September, may be related to the "Senkaku islands conflict" between China and Japan, that indirectly concerns the Philippines.
- Some irregularities may also be explained by major events that disturb the news global behavior. For example, the beginning of the USA presidential campaigns in October may explain why China is much less cited than usual.

In these two experiments, multi-resolution aggregation aims at focusing the analysis on spatial or temporal irregularities. This allows the experts to detect critical events. Agents or groups of agents that are abnormally cited regarding their neighborhood may have an influential role for the newspaper. Broad and narrow peaks in the citations variation may reveal essential events in the country's history. Hence, the aggregation technique presented in [2,3] may be exploited to detect surprising behaviors at different scales of space and time, thus indicating where interesting events can be found for the analysis purposes.

References

1. Grasland, C.: GEOMEDIA: Observatoire des flux géomédiatiques internationaux. Project proposal of ANR-CORPUS (2011)
2. Lamarche-Perrin, R., Demazeau, Y., Vincent, J.-M.: How to Build the Best Macroscopic Description of your Multi-agent System? In: Demazeau, Y., Ishida, T., Corchado, J.M., Bajo, J. (eds.) PAAMS 2013. LNCS, vol. 7879, pp. 157–169. Springer, Heidelberg (2013)
3. Lamarche-Perrin, R., Demazeau, Y., Vincent, J.M.: The Best-partitions Problem: How to Build Meaningful Aggregations? Technical report, Laboratoire d'Informatique de Grenoble, France (forthcoming, 2013)

Demonstrating Multi-layered MAS in Control of Offshore Oil and Gas Production

Lars Lindegaard Mikkelsen[1], Jørgen Rentler Næumann[1], Yves Demazeau[2], and Bo Nørregaard Jørgensen[3]

[1] DONG Energy E&P
{lalmi,jrnae}@dongenergy.dk
[2] CNRS, LIG Laboratory
Yves.Demazeau@imag.fr
[3] University of Southern Denmark
bnj@mmmi.sdu.dk

Abstract. From a control perspective, offshore oil and gas production is very challenging due to the many and potentially conflicting production objectives that arise from the intrinsic complexity of the oil and gas domain. In this paper, we demonstrate how a multi-layered multi-agent system can be used in a satisficing decision-making process for allocation of production resources.

Keywords: Multi-agent systems, Emergence, Satisficing, Multi-objective, Production Systems.

1 Introduction

The background for our research is oil and gas production in the Danish sector of the North Sea, as described in our work [1]. The DONG Energy E&P operated production platform Siri is used as demonstration case. Production software systems have to be revised frequently, as many oil and gas fields are maturing rapidly. Simply applying the same relatively fixed production software systems as of today would result in suboptimal production. The application of a relatively fixed production software system is further challenged by the fact that the growing global request for oil and gas advances technological achievements which allow fields to evolve beyond their original abandonment point.

2 Main Purpose

The main purpose of this demo paper is to demonstrate that the multi-layered multi-agent system proposed in our work [1] dynamically can adapt to new operational conditions. This dynamic control is possible as the infrastructure of the multi-layered multi-agent system takes responsibility for coordinating potential interactions among control agents dynamically. Production objectives and system constraints are represented by agents grouped in negotiation contexts at three decision layers. Each

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 300–303, 2013.

negotiation context holds a mediator agent, which handles the negotiating process in the search for satisficing solutions to the multi-objective production problem. Resource-allocation conflicts may emerge, because agents by default are considered equally important. However, in any non-trivial control system the importance of individual agents may change depending on the actual operational state. This state-dependent change in agents' importance is handled by supporting dynamic prioritization of individual agents. Important agents are given higher priority than less important agents. By default the priority of all agents is set to 5. In the current implementation, we have chosen to use a priority range from 1-10 (1 = highest and 10 = lowest). Fig. 1 depicts to the left examples of agents at the strategic layer, with their priorities in round brackets. To the right is shown the GUI used to change agent priorities at runtime.

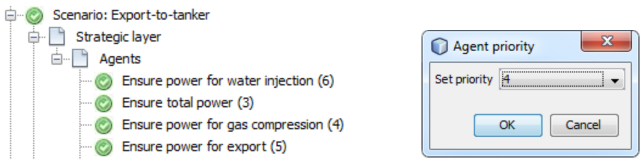


Fig. 1. Agents' priority

3 Experiments

In this demo-paper we use the same export-to-tanker scenario at the Siri platform as described in our work [1], but here with different production configurations using synthetic data. Production configuration refers to availability of production systems and priority of production objectives. The export-to-tanker scenario involves all three decision layers, and is based on oil export to a shuttle tanker from an intermediate storage tank on the seabed. Due to limited electrical power resources under normal operational production conditions at the Siri platform, one of the major power consumers has to be stopped during the export-to-tanker scenario, i.e. either a gas compressor or a water injection pump. The gas compressors are used to handle produced gas either for use as fuel, lift gas (lift gas is used to reduce density of the well fluid to allow the well inflow pressure to overcome the hydrostatic pressure of the fluid column) or re-injection of surplus gas to minimize CO_2 emissions. The gas compressor system consists of three compressors. Water injection is used as pressure support in the reservoirs in order to maintain an economically-feasible production. The water injection system consists of three pumps. Fig. 2 shows the negotiation contexts (solid rectangles) that are directly involved in the experiment.

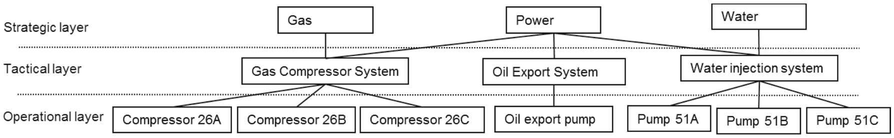


Fig. 2. Experiment negotiation context diagram

The experiment can be split into five phases marked by numbers in Fig. 3 and Fig. 4. Fig. 3 shows the agents' responses to the actual solution, found by the mediator agents, in the form of a fitness value 0-100 [%], where 100% indicates a complete satisficed agent. Fig. 4 shows the water injection system flows [m^3/h].

Experimental preconditions: A) The three water injection pumps are in service with a total capacity of 900 [m^3/h] HP (High Pressure) water to the Siri platform and satellites installations. B) Power to gas compression has lowest priority. C) Export to tanker has not commenced.

1. The export-to-tanker scenario is triggered by the *Oil export system* at the tactical layer, and by requesting power from the *Power* negotiation context. The *Power* mediator agent starts a negotiation process to find a new power plan. The *Power* negotiation context contains the following agents: *Ensure power for water injection*, *Ensure total power*, *Ensure power for gas compression*, *Maximize total power allocated*, and *Ensure power for export*. The new power plan makes, as seen in Fig. 3, the *Ensure power for gas compression* agent respond with a fitness of 83%.
2. The *Ensure power for water injection* agent is given the lowest priority. The *Ensure power for gas compression* is now satisfied, whereas the *Ensure power for water injection* has a fitness of 33%. As seen in Fig. 4, one water injection pump is stopped, i.e. a total of 600 [m^3/h] HP water to the Siri platform and the satellites.
3. The stopped water injection pump is taken out of service for maintenance. The *Ensure power for water injection* has now a fitness of 100%, whereas the *Mixed water request* agent has a fitness of 50%. As seen in Fig. 4, two water injection pumps are still running.
4. The export-to-tanker scenario is completed and the *Maximize total power allocated* agent indicates that a surplus of power is available due to the pump that is out of service for maintenance reasons.
5. The water injection pump that is out of service for maintenance reasons is put back online. As seen in Fig. 4, all water injection pumps are back in operation.

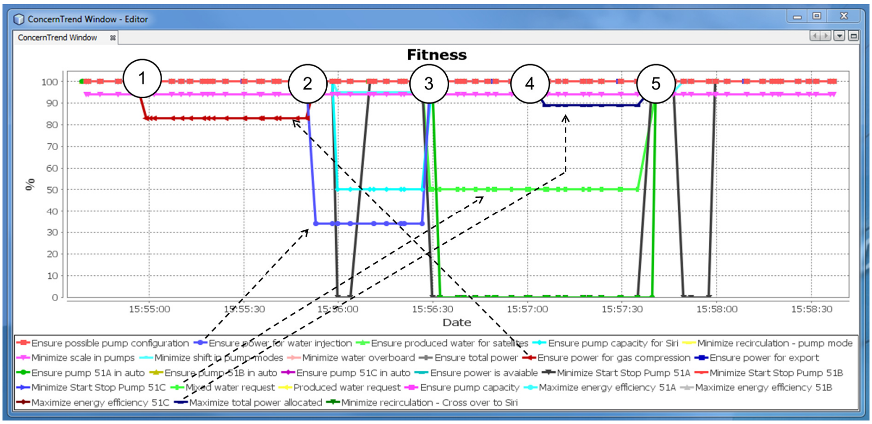


Fig. 3. Agents´ fitnesses

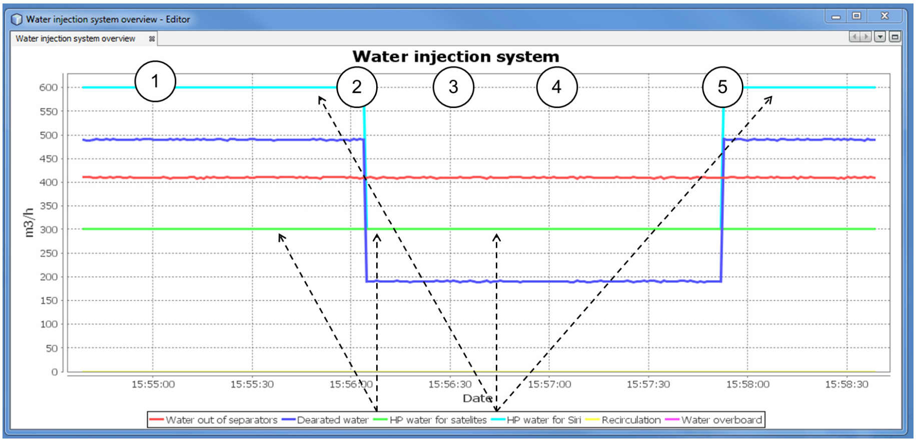


Fig. 4. Water injection system flow [m^3/h]

The trend curves *Water out of separators* and *Dearated water* in Fig. 4 are water supply to the water injection pumps, whereas *Recirculation* is used when ensuring a minimum flow in the pumps and *Water overboard* is used if there is a surplus of *Water out of separators*.

4 Conclusion

We have demonstrated by different production configurations in the same production scenario that a satisficing decision-making process implemented as a multi-layered multi-agent system can handle the changes in full automatic mode. Hence, we believe that the proposed approach [1] possesses the capability to face the continuously changing operational conditions of oil and gas fields. The proposed approach provides a new level of flexibility that meets the need for dynamic evolution of oil and gas fields not seen in the manually controlled systems, as is currently the state of the art within the oil-and-gas-production domain.

Acknowledgement. Noreco and RWE, our partners in the Siri Area, are greatly acknowledged for the permission to publish this paper. The partners do not take responsibility of neither the contents nor the conclusions.

Reference

1. Mikkelsen, L.L., Demazeau, Y., Jørgensen, B.N.: Multi-layered satisficing decision making in oil and gas production platforms. In: Demazeau, Y., Ishida, T., Corchado, J.M., Bajo, J. (eds.) PAAMS 2013. LNCS, vol. 7879, pp. 170–181. Springer, Heidelberg (2013)

IMOSHION: A Simulation Framework Using Virtual Intelligent Agents for Workplace Evacuation in Case of Emergency Situation

Stéphane Maruejouls and Caroline Chopinaud

MASA Group, France
{stephane.maruejouls,caroline.chopinaud}@masagroup.net

Abstract. Every year 5,720 people die in the European Union as a consequence of work-related accidents. To help in reduce the occurrence of incident in case of emergency situation, dedicated tools can be proposed to understand, train and raise people awareness of safety rules. A possible solution is to propose simulation framework or serious games in order to provide people with means to visualize and test their knowledge of safety rules in different context and immersed into their everyday workplace. In this paper, we aim to demonstrate the use of virtual intelligent agents as a solution to design an immersive, adaptive and interactive simulation framework of workplace evacuation.

Keywords: Agent-based simulation, applications of virtual autonomous agents.

1 Introduction

Every year 5,720 people die in the European Union as a consequence of work-related accidents, according to EUROSTAT figures [1]. The International Labour Organisation estimates that 159.500 workers additionally die each year from occupational diseases in the EU [2]. Taking both figures into consideration, it is estimated that every three-and-a-half minutes somebody in the EU dies from work-related causes and that every four-and-a-half seconds an EU worker is involved in an accident that forces him/her to stay at home for at least three working days.

In this context, the IMOSHION[1] (Improving Occupational Safety & Health in European SMEs with help of Simulation and Virtual Reality) project proposes solutions to help SMEs employees, that are the most impacted by safety problems, by developing a set of Occupational Safety & Health (OSH)-related tools such as: OSH knowledge and learning management system [3]; immersive training and learning tool for OSH procedures; planning tool for OSH prevention at workstation and workplace levels; and a workplace simulation for experimenting and training on OSH issues.

This paper focuses on the workplace simulation framework that helps safety service to create representative environment to test procedures during the planning phase

[1] The IMOSHION project is co-financed by the Seventh Framework Programme for Research and Technological Developmement of the European Union.

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 304–307, 2013.

and trains employees and leaders to behave correctly in the case of an emergency situation. For that, the simulation framework uses virtual intelligent agents to provide an adaptive, immersive and interactive environment that is very helpful to train and raise people awareness of, sometime complex, safety rules.

In [4] scenarios-based agent systems are used to offer excellent promise in simulating crowd evacuation using real-world experiment. Agent-based simulations are also used to test technologies or situation that not easily tested in real environment involving disaster or incident. For instance DrillSim [5], based on multiagent simulation, simulates disaster response activity by modelling each people as an agent. Following these approaches, IMOSHION aims to propose a complete easy to use simulation framework and focuses on demonstrate the contribution of agent-oriented approaches to design new adaptive scenarios of a simulation.

2 Main Purpose

The current situation in relation to OSH is that all SMEs in Europe have to obey the national derivatives of the European Framework Directive (89/391/EEG) and its underlying twenty-three guidelines. As there are many more rules to follow, this is felt by the SMEs primarily as a huge administrative burden. One obligation according to the European Directive is the risk inventory that the companies should perform. For the SMEs this is a recurrent, time-consuming and complex procedure leading only to the definition of risks, while the finding of concrete measures to eliminate these risks requires another time-consuming search process.

The aim of the simulation framework is to propose a virtual environment that allows understanding the right and inappropriate behaviours to follow in case of evacuation of a workplace when an incident occurs, such as fire, explosion and toxic leak. Moreover, because real emergency exercises are costly in time and money and emergency situations are hard to initiate in reality, the use of a simulation framework, which allows people to get proper training, is a powerful solution upon condition that the simulation is user-friendly and proposes realistic situation.

Thus, the main challenge of IMOSHION simulation framework is to simulate as much as possible realistic human behaviors, especially in terms of decisional aspects regarding the situation evolution and interactions. Thus, agents claim to behave as autonomous, adaptive and situated agents.

The simulation framework proposed two kinds of execution: a third-person view, in which only virtual agents are evolved by following autonomous behaviours, and a first-person view in which a user manipulates an avatar to interact with the environment and the virtual agents.

In both usages, to design relevant scenarios, the simulation framework provides a tool to quickly build a dedicated and representative environment, and a library of human behaviours composed by common workers' activities, following safety rules behaviours and inappropriate behaviours. One of the main advantages of the use of adaptive, interactive and autonomous agent to construct a scenario is to permit to continuously propose new relevant scenarios in which the situation should evolved differently depending on the user's interaction.

3 Demonstration

We illustrate the relevance of the use of agents in the simulation framework along two lines: the first one demonstrates how the simulation can be used to visualize the safety of an environment and understand right and wrong behaviours, the second one demonstrates how to use the simulation to train leaders and employees to safety evacuation rules in case of emergency situation.

In the simulation, the agents are designed and developed through the AI-middleware MASA LIFE that provides means to model, execute and integrate agent behaviors using the Behavior Tree algorithm [6]. The simulated environment and the agents rendering are done through an integration of the decisional behaviors into the Unity3D engine with RocketBox animation.

Step1. We show how to use the *Building Editor* of the simulation framework to create a 3D-representation of workplace building in a quick and user-friendly way.

Step2. The *Exercice Editor* allows to create an emergency scenario in three phases: (i) Place office furniture such as desks, chairs, industrial tools and safety furniture such as alarms, extinguisher, fire doors and safety zone (ii) Choice, schedule and locate the incident (iii) Populate the environment with various decisional agents with specific or common behaviours from the provided library.

Step3. The simulation is run over the exercice showing agents with rich and realistic behaviours in terms of decisional aspects such as:

- "Leader Agent" which is assigned to evacuate the building by checking every room, potentially triggering the alarm, using extinguisher, closing fire doors and finally helping others agents to leave the building. These agents adapt theirs behaviours according to the situation and strongly interact with the environment.
- "Trained Agent" which knows evacuation process and move to its assigned safety zone or the closest one in respect of safety rules. These agents are designed to show right behaviours in case of incident.
- "Panicked Agent" which freezes in high stress or run around without close fire doors. These agents are especially designed to show wrong and irrelevant behaviours.
- "Injured Agent" which represents people who walk slower or might be lost in the building. Leader or trained agents should have to interact with this kind of agent in order to help it to go to the safety zone.

Step4. The simulation is run successively in the two modes. Firstly with the third-person view, we can show the execution of the scenarios and the correct location of the furniture in the building thanks to the execution of the agents and how the agents interact with the environment. Secondly with the first-person view, we can demonstrate how the agents can adapt their behaviours to the presence of the user's avatar. At the end of the training mode, the simulation credits the user with an evaluation score. This score is used to evaluate the user and track his progression and awareness of emergency procedures.

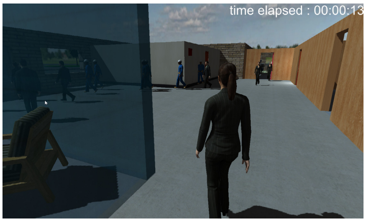


Fig. 1. First-person training view

Fig. 2. Third-person test view

4 Conclusions

The IMOSHION simulation framework uses autonomous virtual agents to train employees to behave correctly in case of emergency situation. We demonstrate the interest of agent approaches to make scenarios evolved according to users interaction and increase the presence of the user in the simulation. We promote the use of intelligent agents into simulation and serious games in order to construct more relevant and immersive scenarios, or to evaluate and test environment architecture and structure.

Future works concern the improvement of the interactive behaviours of the agents in the simulation and find solutions to provide the design safety services to directly construct very new scenarios and behaviours. The main goal of our AI-middleware is to easily design decisional behaviours through an accessible graphical behaviours editor and a sandbox to quickly test the behaviours. So, associated with the simulation framework, the AI-middleware should allow building new scenarios through the design of new behaviours of virtual agents and then proposes unlimited test and training possibilities.

References

1. EUROSTAT: 8.6% of workers in the EU experienced work-related health problems. Statistics Focus 63/2009. Population and Social Conditions (2009) ISSN 1977-0316
2. International Labour Organisation: Occupational Safety and Health. Document (2010)
3. Shalloe, S., Lawson, G., D'Cruz, M., Eastgate, R.: User-centered development of an occupational health and safety knowledge and learning management system. Poster in Applied Human Factors and Ergonomics (AHFE) Conference, San Francisco (2012).
4. Murakami, Y., Minami, K., Kawase, T., Ishida, T.: Multi-agent Simulation for Crisis Mangement. In: IEEE Workshop on Knowledge Media Networking, Japan (2010)
5. Massaguer, D., Balasubramanian, V., Mehrotra, S., Venkatasubramanian, N.: Multi-Agent Simulation of Disaster Response. In: Proceedings of Autonomous Agents and Multi-Agent Systems conference (AAMAS 2006), Hokodate, Japan (2006)
6. Millington, I., Funge, J.: Artificial Intelligence for Games, 2nd edn., ch. 5, section 5.4 Behavior Trees. Focal Press (2009)

A Security Response Approach Based on the Deployment of Mobile Agents: A Practical Vision

Roberto Magán-Carrión, José Camacho-Páez, and Pedro García-Teodoro

Department of Signal Theory, Telematics and Communications,
Faculty of Computer Science - CITIC, University of Granada
{rmagan,josecamacho,pgteodor}@ugr.es

Abstract. The existence of malicious nodes in mobile ad hoc networks (MANET) can degrade (and even interrupt) the overall operation of the network. To avoid this we propose a novelty multi-agent response/tolerance scheme based on the dynamic deployment of distributed agents to sustain the quality of the service in the presence of malicious nodes. The optimal locations of the agents are computed from the particle swarm optimization (PSO) algorithm, which maximizes the global network connectivity and thus mitigates the negative effects produced by malicious nodes on the network performance. Since this paper tries to provide a practical vision of the system developed, we focus here on the associated functional architecture and its operation in a simulated scenario.

Keywords: agent, PSO, detection, malicious, MANET, response, survivability, tolerance.

1 Introduction

MANETs has become a new communication paradigm for the last years due to some of their interesting inherent features. However, the same characteristics make this kind of environments highly vulnerable to several types of security threats. For example, specific attacks such as *dropping* or *selfish* [1] have a very high impact over the network performance in multihop routing environments. Therefore, the deployment of security mechanisms to strengthen the services provided and thus sustain the network performance in the presence of attacks is recommended. This way, implementing effective attack detection (recognition) and response/tolerance (recovery) mechanisms constitutes a key issue for network survivability [2].

To the best of our knowledge, there are a low number of works in the specialized literature where a multi-agent based solution is used to detect and/or react against security attacks in MANETs. Moreover, the few solutions currently available mainly refer to software oriented solutions. In this line, immune agent based solutions are addressed in the literature [3]. Some other proposals are tolerance

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 308–311, 2013.

related approaches, as that in [4], where a re-routing function is performed over the network through the assignment of trust values to the monitored nodes.

In the present work we propose to implement a response/tolerance security solution by deploying physical agent nodes to solve the malicious consequences of *packet dropping* behaviors in MANETs. The scheme is aimed at restoring the function avoided by the malicious nodes (*droppers*) by dynamically moving the agents to the optimal positions to maintain the connectivity of the overall network over time.

2 Main Purpose

We have developed a response approach for improving system survivability against security threats in MANETs, in particular, to react against common malicious behaviors such as selfish or dropping. The system is composed of two main types of nodes: *user nodes* (UN), and *agent nodes* (AN). The first ones, UNs, refer to usual legitimate MANET nodes involved in normal network services. Instead, ANs are agents dynamically deployed by the administrator in the environment to guarantee that the UNs receive the required network service. Additionally to UNs and ANs, our study also involves the so-called *malicious nodes* (MN), which present harmful behaviors for the overall network performance. In our experimentation the MNs are able to behave over time both as normal user nodes as well as damaging nodes.

Our proposal is based on the previous work developed by Dengiz et al. in [5], where the authors make use of the *particle swarm optimization* (PSO) algorithm together with a motion prediction procedure for mobile UNs location over time. This way, the agents are re-located dynamically to the optimum positions obtained by the PSO algorithm from: (a) the motion prediction of the UNs for a next time instant $t + H$ through a kinematic prediction procedure, and (b) the evaluation of three optimization functions related to the connectivity and the flow transmitted through the network. The output of the PSO algorithm is the set of optimal positions to be occupied by the ANs throughout the network in the next instant aiming the network connectivity maximization.

3 Demonstration: MARS (Mobile Agent Response System)

Our specific tolerance approach is named MARS, from Mobile Agent Response System. For MARS functioning as a response system, a previous detection module is needed to monitor the environment, detect the potential occurrence of malicious dropping behaviors and trigger alarms if necessary. Such detection module is out of the scope of this work and thus it is has not been developed here. An example of this kind of systems can be found in [6].

Once a dropper node is detected, the MARS module is executed to avoid the harmful effects of the attack. This way, a set of mobile agents are dynamically

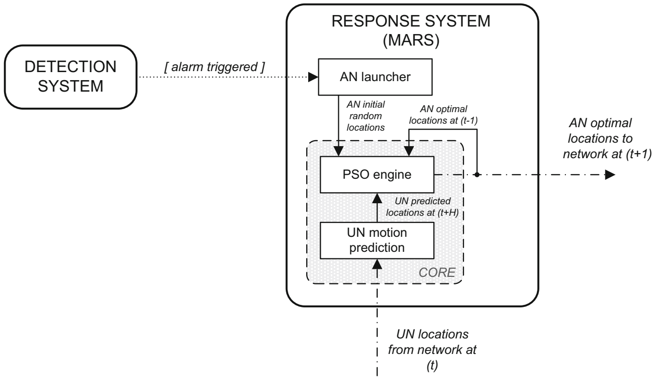
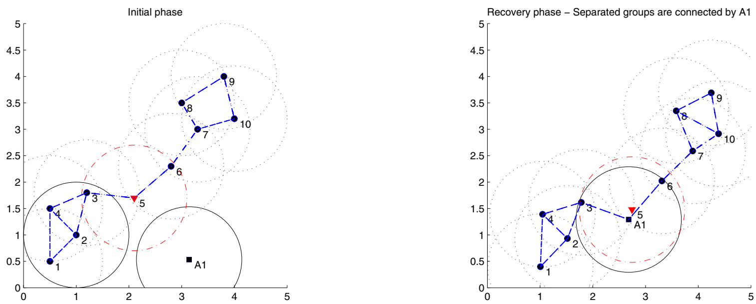


Fig. 1. Functional architecture for MARS

deployed in the environment, with two main features. First, they act as relaying nodes to solve the decrease in connectivity in the network due to the appearance of MNs. Second, the base optimization algorithm proposed in [5] and described in the previous section is performed to determine the best future positions of the ANs to maximize the overall connectivity of the network. Figure 1 depicts the functional architecture of MARS and how it operates in presence of attacks. Regarding the core module of the system, it is continuously working while the attack remains active. This way, the PSO engine makes use of the last optimal AN positions at $t-1$ and the predicted locations of UNs at $t+H$ to obtain the new optimal positions of ANs at $t+1$, t being the current execution time. Therefore, the ANs move to the closer positions for which, in our case, the network connectivity is maximized.



(a) (b)

Fig. 2. Connectivity maximization and recovery process provided by MARS. (a) Initial phase where a malicious node (inverted triangle) is performing the expected forwarding process together with the legitimate user nodes (solid circles). Then two separated groups of nodes result when the attack is carried out. (b) The agent A1 is approaching to its optimal location to recover the connectivity and thus overall communications.

Figure 2 illustrate how MARS actually works. Initially, several UNs are randomly distributed throughout a given network area, where there is one MN (node 5) and one AN (node A1). At the beginning, in Figure 2(a), the malicious node works as a normal node. Afterwards, the attack is produced, so that the overall connectivity is broken and two separated networks appear. Few time steps later, the agent A1 is approaching to its optimal position with the aim to recover the connectivity lost. This is shown in Figure 2(b).

4 Conclusions

In this work we have introduced a novel response/tolerance mechanism based on the use of physical multi-agents. A positioning optimization procedure is carried out to determine the best positions of the agents over time to dynamically maximize the connectivity of our communication environment. We test our system by means of a simulated scenario, from which it can be concluded the promising nature of our approach. The proposed scheme is able to recover and maintain the connectivity lost due to the appearance of malicious nodes that do not cooperate in the network forwarding process.

Acknowledgment. This work has been partially supported by Spanish MICINN through project TEC2011-22579 and by the FPU P6A grants program of the University of Granada.

References

1. Ehsan, H., Khan, F.A.: Malicious AODV: Implementation and Analysis of Routing Attacks in MANETs. In: Procedings 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2012), UK, pp. 1181–1187 (June 2012)
2. Lima, M., dos Santos, A., Pujolle, G.: A survey of survivability in mobile ad hoc networks. IEEE Communications Surveys & Tutorials 11, 66–77 (2009)
3. Ye, X., Li, J.: A security architecture based on immune agents for MANET. In: International Conference on Wireless Communication and Sensor Computing, ICWCSC 2010, pp. 1–5. IEEE (2010)
4. Halim, I.T.A., Fahmy, H.M.A., Bahaa El-Din, A.M., El-Shafey, M.H.: Agent-Based Trusted On-Demand Routing Protocol for Mobile Ad Hoc Networks. In: 2010 4th International Conference on Network and System Security (NSS), pp. 255–262. IEEE (2010)
5. Dengiz, O., Konak, A., Smith, A.E.: Connectivity management in mobile ad hoc networks using particle swarm optimization. Ad Hoc Networks 9, 1312–1326 (2011)
6. Sánchez-Casado, L., Maciá-Fernández, G., García-Teodoro, P.: An Efficient Cross-Layer Approach for Malicious Packet Dropping Detection in MANETs. In: Procedings 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2012), UK, pp. 231–238 (June 2012)

The Galaxian Project: A 3D Interaction-Based Animation Engine

Philippe Mathieu and Sébastien Picault

Université Lille 1, Computer Science Dept. LIFL (UMR CNRS 8022),
Cité Scientifique 59655 Villeneuve d'Ascq Cedex, France
firstname.surname@univ-lille1.fr
http://www.lifl.fr/SMAC/

Abstract. This demonstration paper presents a space battle simulation, which was designed using the Interaction-Oriented approach IODA, and implemented within the 3D professional game engine Unity. After giving an overview of the simulation, we explain how it was built through a step-by-step setup, and how the use of interactions enable infinite extensions.

Keywords: Agent-based simulation, Interaction-Oriented Design, Video Games.

1 Introduction

The Galaxian Project [1] was developed in 2011–2012 among the Multiagent research team (SMAC) at Lille 1 University, France. It was designed to run for hours on the 6 m-wide screen of the PIRVI platform[1] at the Computer Science Laboratory (LIFL).

It aims at demonstrating the capabilities of the design principles and algorithms we promote through the Interaction-Oriented approach, the IODA method [2,3], especially for video game applications. Actually, this simulation presents an application of the IODA method to a 3D space battle, involving several kinds of spacecrafts. It is implemented within a commercial video game engine (Unity 3D[2]), with embedded pedagogical material.

2 Main Purpose

Two teams are fighting: white against black. The simulation relies upon the following agents:

- the white **fighters** are the atomic units of the white team, and assault the black fighters – and vice-versa;
- the white **crusers** (resp. black) send new white fighters from time to time so as to revive the battle (actually they do nothing more);

[1] "Plateforme Interactions-Réalité Virtuelle-Images": http://www.lifl.fr/pirvi/
[2] http://unity3d.com

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 312–315, 2013.

- **frigates** are found only in the black team: though smaller than the crusers, they are strong ships, armed with powerful turrets, and are able to fight off several white fighters at a time;
- **squads** are built only by white fighters which decide to join together, in order to be strong enough to attack black frigates, then they break up.

Galaxian relies upon the IODA method, therefore the behaviors of all those agents are expressed as *interactions*, i.e. conditions-actions rules involving two agents (a *source* agent which can perform the interaction, and a *target* which can undergo it). In addition, thanks to the modularity of this method, the behaviors were designed and tuned incrementally.

Since the simulation is aimed at running for hours, the navigation interactions were tuned first (SeekTarget, Confront, Engage, Escape) so as to ensure that spaceships keep moving in front on the camera. Then, struggle behaviors were implemented (Fire, Explode, LaunchFighter, etc.) and tested. Finally, team behavior such as creating, joining and disbanding squads were added. Additional agents and behaviors could be easily introduced in the same way to extend the simulation.

The resulting simulation model is summarized in the *interaction matrix* below (for more details about this approach, please refer to [3]):

Table 1. This *interaction matrix* describes which interactions are assigned to source/target agent families. For instance, the intersection of line “WFighter” and column “BFighter” contains all interactions that a white fighter agent can perform on a black fighter agent. Each interaction is followed by a priority level (from the point of view of the source agent) and a limit distance (which constrains the distance between source and target to make the interaction possible). The “∅” column contains *reflexive* interactions (where the target is the source itself).

Targets / Sources	∅	BFighter	WFighter	BFrigate	WSquad
BFighter	(SeekTarget; 0) (Fire; 8) (Explode; 9)		(Confront; 2; 1000) (Intercept; 3; 1000) (Engage; 4; 1000) (Escape; 5; 1000)		
WFighter	(SeekTarget; 0) (Fire; 8) (Explode; 9)	(Confront; 2; 1000) (Intercept; 3; 1000) (Engage; 4; 1000) (Escape; 5; 1000)	(CreateSquad; 2; 30)		(Join; 6; 100) (Follow; 7; 1000)
BFrigate					(FightBack; 0; 200)
BCruser	(LaunchFighter; 0)				
WCruser	(LaunchFighter; 0)				
WSquad	(Disband; 0) (Fire; 3)			(Engage; 1; 5000)	(Merge; 5; 50)

The original IODA simulation engine, called JEDI, was developed in Java [3]. The main transformations to integrate it within Unity were the following:

- a rewriting from Java to C#
- a specific scheduler, so as to let the Unity engine deal with 3D computations, collisions, and rendering at its own variable framerate, while the IODA engine is in charge of making agents select their behavior at fixed time steps, according to their perceived neighbors, their state and the interaction matrix.

3 Demonstration

The simulation is able to run a battle for hours. In order to visualize which behaviors occur during the simulation, the simulation is endowed with gizmos showing:

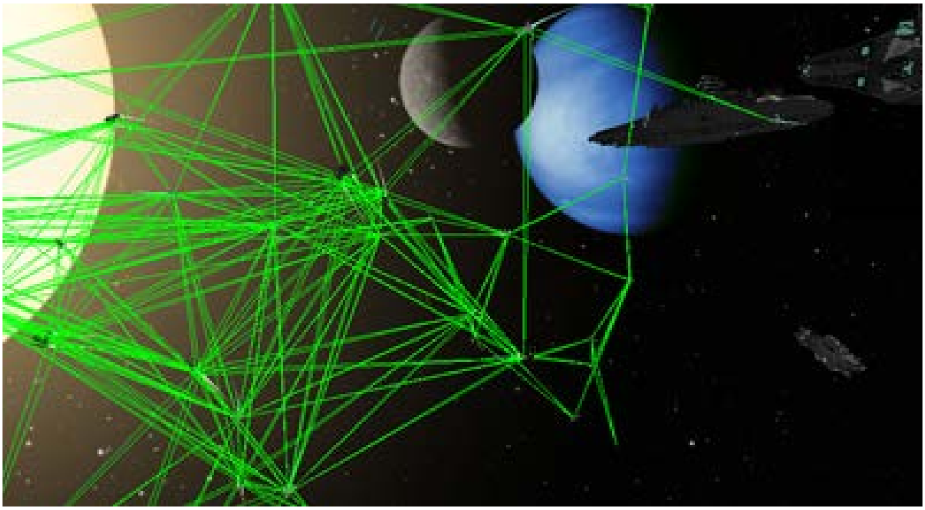


Fig. 1. A view of the Galaxian simulation. The green lines between agents show their perceptions.

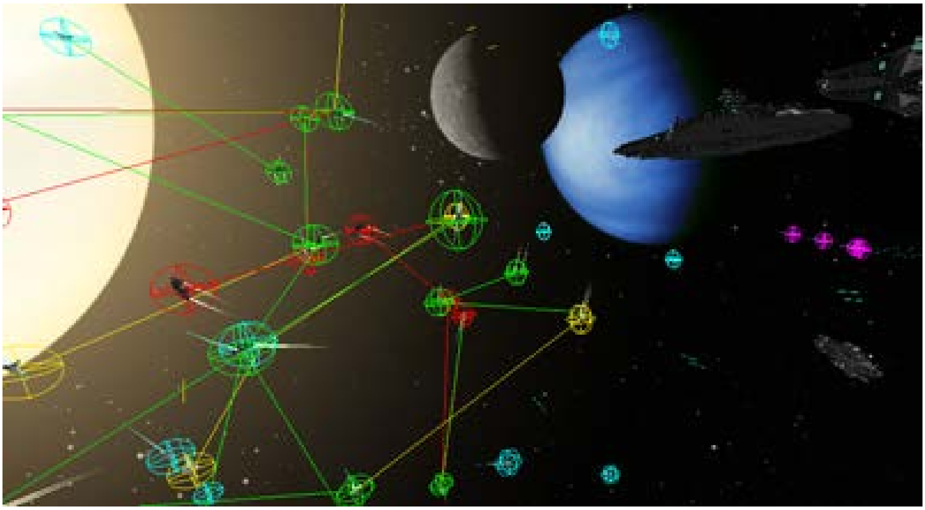


Fig. 2. A view of the Galaxian simulation. The colored circles surrounding the agents represent the interaction they are currently performing and the line ends on the corresponding target agents (the agent that is undergoing this interaction).

– the perception links between agents (see fig. 1)
– the interactions that are performed by agents (as sources), with a link to the corresponding target (see fig. 2).

Due to the portability of the Unity models, a web version is provided on-line [1]. In addition, the full demonstration (stand-alone application) encapsulates slides to explain how the IODA simulation engine selects the appropriate interactions in several concrete cases.

4 Conclusion

This demonstration shows how complex simulations, e.g. in video games, can easily be designed through an incremental process with the IODA approach. Any model can be extended simply by writing new interactions and assigning them to agents in a declarative structure, the interaction matrix. Besides, this method provides a homogeneous representations of all entities (as agents) and all behaviors (as interactions).

Acknowledgments. The Galaxian project was implemented by Marc-Antoine Dupré (associate engineer, supported by the French research institute INRIA), under the supervision of David Panzoli (post-doctoral researcher).

References

1. Dupré, M.A., Mathieu, P., Panzoli, D., Picault, S.: Galaxian, http://www.lifl.fr/SMAC/projects/galaxian
2. Kubera, Y., Mathieu, P., Picault, S.: Interaction-Oriented Agent Simulations: From Theory to Implementation. In: Proceedings of the 18th European Conference on Artificial Intelligence (ECAI 2008), pp. 383–387 (2008)
3. Kubera, Y., Mathieu, P., Picault, S.: IODA: an interaction-oriented approach for multi-agent based simulations. Journal of Autonomous Agents and Multi-Agent Systems, 1–41 (2011)

Demonstration of the Multi-Agent Simulator of Competitive Electricity Markets

Tiago Pinto, Isabel Praça, Gabriel Santos, and Zita Vale

GECAD – Knowledge Engineering and Decision-Support Research Center,
Institute of Engineering – Politechnic of Porto (ISEP/IPP), Porto, Portugal
{tmcfp,icp,gajls,zav}@isep.ipp.pt

Abstract. Electricity markets are complex environments with very particular characteristics. A critical issue concerns the constant changes they are subject to. This is a result of the electricity markets' restructuring, performed so that the competitiveness could be increased, but with exponential implications in the increase of the complexity and unpredictability in those markets' scope. The constant growth in markets unpredictability resulted in an amplified need for market intervenient entities in foreseeing market behavior. The need for understanding the market mechanisms and how the involved players' interaction affects the outcomes of the markets, contributed to the growth of usage of simulation tools. Multi-agent based software is particularly well fitted to analyze dynamic and adaptive systems with complex interactions among its constituents, such as electricity markets. This paper presents the Multi-Agent System for Competitive Electricity Markets (MASCEM) – a simulator based on multi-agent technology that provides a realistic platform to simulate electricity markets, the numerous negotiation opportunities and the participating entities.

Keywords: Adaptive learning, electricity markets, multi-agent simulation.

1 Introduction

Electricity markets worldwide are complex and challenging environments, involving a considerable number of participating entities, operating dynamically trying to obtain the best possible advantages and profits [1]. The recent restructuring of these markets increased the competitiveness of this sector, leading to relevant changes and new problems to be addressed [1, 2]. Potential benefits depend on the efficient operation of the market [1]. Market players and regulators are very interested in foreseeing market behavior, thus a clear understanding of the impact of power systems physics on market dynamics and vice-versa is required. It is essential to fully understand the market's principles and learn how to evaluate investments in such environment [2].

The development of simulation platforms based in multi-agent systems is increasing to simulate real systems in which stakeholders have different and often conflicting objectives. These systems allow simulating scenarios and strategies, providing users with decision making according to their profile of activity. The use of multi-agent systems for the simulation of electricity markets is a reality [3, 4, 5] as are examples the AMES [3] and EMCAS [4] simulators.

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 316–319, 2013.

MASCEM - Multi-Agent Simulator for Electricity Markets [5, 6, 7] is also a modeling tool to study and explore restructured electricity markets. Market players are complex entities with their own characteristics and objectives, using adequate strategies in order to achieve their goals. MASCEM multi-agent model includes players with dynamic strategies, acting in several types of markets, using real electricity markets' data to base the simulations.

In real electricity markets, a player's behavior depends on the self-knowledge that it detains. It changes depending on the new information and knowledge that the player may obtain from the exterior and from the dynamic complex interactions with other heterogeneous entities. MASCEM players are endowed with learning processes, though the connection with another multi-agent system: ALBidS (Adaptive Learning strategic Bidding System) [6]. ALBidS provides decision support to electricity markets' negotiating players, allowing them to analyze different contexts of negotiation, and automatically adapt their strategic behavior according to the current situation. This system implements several negotiation mechanisms and data analysis algorithms, enhancing the strategic behavior of the players.

2 Main Purpose

MASCEM's purpose is to be able to simulate as many market models and player types as possible, enabling it to be used as a simulation and decision-support tool for short/medium term purposes but also as a tool to support long-term decisions, such as the ones taken by regulators. Taking into account the realities of several countries with different perspectives and approaches in what concerns the electricity negotiation, MASCEM allows the simulation of the main market models found worldwide [8]: day-ahead pool (asymmetric or symmetric, with or without complex conditions), bilateral contracts, balancing market, forward markets and ancillary services. It also allows hybrid simulations, regarding players operation in a chosen combination of the mentioned market models. This implies that each agent must decide whether to, and how to, participate in each market type.

Players in MASCEM are another main concern, since the operation of electricity markets in hugely dependant on how players act and interact in this environment. The study of the interactions between market players, and their decisions when negotiating is as important as the modeling of the market mechanisms themselves. For that MASCEM considers all the most important entities, with their own decision-support resources, and allowing the definition of their offers and strategies, granting them competitive advantage in the market. It modulates the complexity of dynamic market players, their interaction and medium/long-term gathering of information (data and experience in the market). MASCEM main entities include: a market operator agent, a system operator agent, a market facilitator agent, buyer agents, seller agents, Virtual Power Player (VPP) [7] agents, and VPP facilitators.

VPPs [7] represent alliances of players, mainly smaller players, with small or null possibility to face negotiations with much more powerful entities, providing the means to adequately support their participation in the competitive electricity markets. They manage the information of their aggregates and are viewed from the market as common seller or buyer agents. VPP agents are implemented as a coalition of agents, each one acting as an independent multi-agent system, maintaining high performance and allowing agents to be installed on separate machines.

Another of the main purposes of this simulator is providing decision support to these entities. For that MASCEM is integrated with another multi-agent system: ALBidS (Adaptive Learning strategic Bidding System) [6], which uses several distinct technologies and approaches to support players' actions in the market.

ALBidS uses reinforcement learning algorithms to choose the player's most adequate action, from a set of different proposals provided by the several algorithms that present distinct approaches. To this end, the reinforcement learning algorithm considers the past experience of the action's responses and the present characteristics of each situation, such as the week day, the period, and the specific market types.

3 Demonstration

In order to provide an adequate and useful framework for electricity markets simulation, MASCEM provides a rather user friendly interface, allowing users to define all the simulation characteristics. The user selects the market types and players to simulate, and all the associated characteristics. These data can also be provided by Excel files, to facilitate the input of data. Important and valuable simulations can be saved to be re-run later, or compared with other simulations with alternative definitions of market models, players' characteristics or players' action profiles.

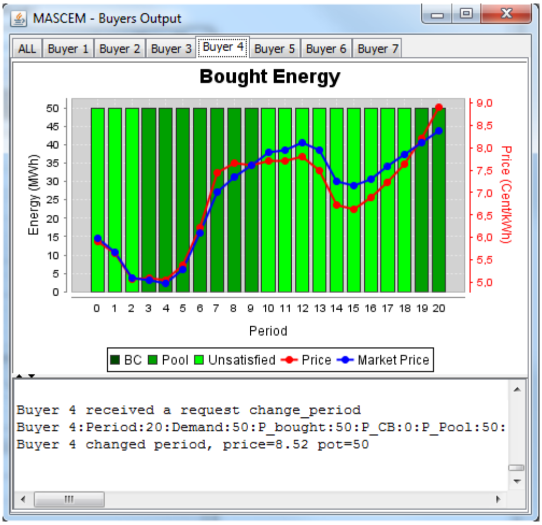


Fig. 1. Buyer agent's output in a spot market simulation using MASCEM [6]

The presentation of results is done through graphical charts, containing amounts of energy sold or bought in each market, the comparison between market prices and prices resulting from players actions in such markets, the profits or costs that each player achieved in the market actions, and the comparison between all players' proposed prices and the actual market price that was achieved. Fig. 1 presents the output of a buyer agent when acting in the day-ahead spot market (auction based).

The top part of Fig. 1 shows the graphical representation of this agent's results in the pool. It includes the amount of energy that it bought in each period, and the comparison between its bid and the market price. The bottom part presents requests that this agent is receiving, and some information about this agent's actions. In this case period 20 has ended and the agent is preparing to start negotiations of period 21.

4 Conclusions

The use of multi-agent systems for the simulation and study of competitive energy markets is increasing as a great asset in the area of power systems. MASCEM is a valuable tool for various targets. Professionals, namely operators and regulators can use it foresee market behaviors and test the adequacy of market mechanisms in simulations using real data from electricity markets. They can also use MASCEM simulations to test new solutions using simulated scenarios, without influencing the real world through tests with possibly catastrophically outcomes.

Real electricity market players can use the decision support capabilities of MASCEM to experiment behaviors and actions with the goal of achieving the best possible outcomes. Also, players can use the real data based simulations of MASCEM to understand competitor players' behaviors, granting them competitive advantage.

Trainees and power systems' area students are provided with the chance to study and understand electricity markets' operation, players' interactions, and this environment as a whole. Finally, researchers are gifted with a powerful tool to deeply and thoroughly study the characteristics and particularities that arise from the interactions, mechanisms and behaviors present in electricity markets, and improve the current solutions to face such a competitive, dynamic, and important environment.

Acknowledgments. This work is supported by FEDER Funds through COMPETE program and by National Funds through FCT under the projects FCOMP-01-0124-FEDER: PEst-OE/EEI/UI0760/2011, PTDC/EEA-EEL/099832/2008, PTDC/SEN-ENR/099844/2008, and PTDC/SEN-ENR/122174/2010.

References

1. Shahidehpour, M., et al.: Market Operations in Electric Power Systems: Forecasting, Scheduling, and Risk Management, pp. 233–274. Wiley-IEEE Press (2002)
2. Meeus, L., et al.: Development of the Internal Electricity Market in Europe. The Electricity Journal 18(6), 25–35 (2005)
3. Li, H., Tesfatsion, L.: Development of Open Source Software for Power Market Research: The AMES Test Bed. Journal of Energy Markets 2(2), 111–128 (2009)
4. Koritarov, V.: Real-World Market Representation with Agents: Modeling the Electricity Market as a Complex Adaptive System with an Agent-Based Approach. IEEE Power & Energy Magazine, 39–46 (2004)
5. Praça, I., et al.: MASCEM: A Multi-Agent System that Simulates Competitive Electricity Markets. IEEE Intelligent Systems 18(6), 54–60 (2003)
6. Vale, Z., et al.: MASCEM - Electricity markets simulation with strategically acting players. IEEE Intelligent Systems 26(2) (2011)
7. Pinto, T., et al.: A new approach for multi-agent coalition formation and management in the scope of electricity markets. Energy 36(8), 5004–5015 (2011)
8. OMIE – Operador del Mercado Iberico de Energia website, http://www.omie.es/ (acessed on January 2013)

Multi-Agent Systems Platform for Mobile Robots Collision Avoidance

Angel Soriano, Enrique J. Bernabeu, Angel Valera, and Marina Vallés

Instituto U. de Automática e Informática Industrial
Universitat Politècnica de Valencia,
Camino de Vera s/n 46022 Valencia, Spain
{ansovi,ebernabe,giuprog,mvalles}@ai2.upv.es

Abstract. This paper presents a multi-agent platform to simulate a new methodical approach to the problem of collision avoidance of mobile robots, taking advantages of multi-agents systems to deliver solutions that benefit the whole system. The proposed method has the next phases: collision detection, obstacle identification, negotiation and collision avoidance. In addition of simulations with virtual robots, in order to validate the proposed algorithm, an implementation with real mobile robots has been developed.

Keywords: Avoiding collision method, robotic agents, mobile robots, avoidance collision method, jade platform.

1 Introduction

As it is well known, the main theme of AI is the concept of intelligent agent defined as an autonomous entity which observes through sensors and acts upon an environment using actuators [9]. This definition is very close to features that a robot can provide, so the concept of agent often is related with robots [3], [14].

On the other hand, detecting and avoiding a collision is a previous step for overcoming the motion planning problem in mobile robots. Collision-detection techniques can be divided into discrete collision detection (DCD), and continuous collision detection (CCD). The DCD algorithms involve stepping the motion of both the mobile robot and the mobile obstacle at a time sampling rate and some problems like tunneling can occur [10]. The CCD techniques are more effective because motions are not stepped. CCD algorithms basically make a return if a collision between the motion of two given objects is presented or not; and if it is going to occur then, the instant in time of the first contact is returned [5], [12] and [1].

In this paper, a multi-agent systems platform for collision avoidance for mobile robots is proposed. Strategies of collision detection of autonomous mobile robots based on [2] are combined with strategies based on artificial intelligence to offer a new method of avoiding collision management. The platform for the implementation and management of the multi-agent system is based on JADE [6].

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 320–323, 2013.

2 Main Purpose

This platform expects to present a new methodical approach to the problem of collision avoidance of mobile robots taking advantages of multi-agents systems (MAS) to deliver solutions that benefit the whole system. The method is divided into three basic concepts: obstacle detection by a mobile robot, the concept of abstraction robotic agent as a software agent within MAS, and distributed artificial intelligence as a method of communication and negotiation between these software agents.

The aim of the avoiding collision method used in this platform is to implement a methodology for obtaining the instant in time when two robots or agents in motion will be located at their maximum-approach positions while they are following straight-line trajectories. If the involved robots do not collide then their minimum separation is returned. Otherwise, their maximum penetration is computed as a minimum translational distance [4]. A very remarkable aspect is both the instant in time and the corresponding minimum separation or maximum approach are computed without stepping any involved trajectory.

The methodology used for the collision avoidance platform has the next phases:

Detection: In this phase, the local system of the robot detects an obstacle that may be a threat of collision at some point and calculates the position of threat-object in the global scenario. This position is sent to the agent who represents the local system in MAS to manage the threat.

Obstacle identification: When an agent receives a position of a threat-object by the local system it represents must identify what kind of threat it is, a moving object or a static object. In order to do this, the agent consults the other agents to know who is located within that area of threat. If there isn't any agent within that area, the threat is identified as a threat of static object.

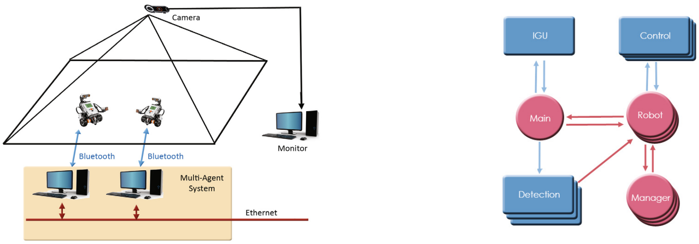
Time to talk, negotiate and resolve: When the two involved agents in a possible threat have been identified, the communication between the two agents is used to obtain the information needed to apply the detection algorithm.

Collision detection algorithm application: This algorithm requires the current position coordinates of detector-agent and threat-object and their destinations, and it returns the probability of collision. In case of collision, the method informs to detector-agent the time of maximum penetration to be produced.

Negotiation: To decide the load percentage that each robot will have in the collision avoidance, the two agents communicate with each other and exchange parameters such like priority, the weight of the transported load, the difficulty of maneuvering, speed… This information defines the easiness or availability that each agent offers to change its trajectory and avoid collision.

Solve the collision: The detector-agent computes the two new positions that the robot should be achieve at correct time to avoid collision. The threat-agent receives the avoidance position and the time in which must be achieve. Once it's reached, the collision is resolved, and each robot continues its original path.

The platform allows simulated and real robots. The real robots are based on the LEGO Mindstorms NXT systems [7] programmed with LeJOS [8] and equipped with four SHARP IR proximity sensors [11] connected by a I2C bus. The robots are connected to their software agents (computers) via Bluetooth and those computers are part of a network that forms the overall MAS through JADE. Figure 1 (a) shows the platform control architecture.



(a) (b)

Fig. 1. Platform Control architecture

In Fig. 1 (b), purple circles represent agents, and blue rectangles represent execution threads. Each robot has a control module for following trajectory, a detection module that manages the IR sensors and a manage module that obtain the data and solve the collisions.

Figures 2 and 3 shows two executions obtained with the proposed platform. In the first one four robots (circles green, blue, black and pink) are considered. The robot initial positions are the corners of the arena, and they must arrive to the opposite corner (marked by a star). The figure also shows the detection area (a trapezoid in front of each robot), and the final path described by the robot.

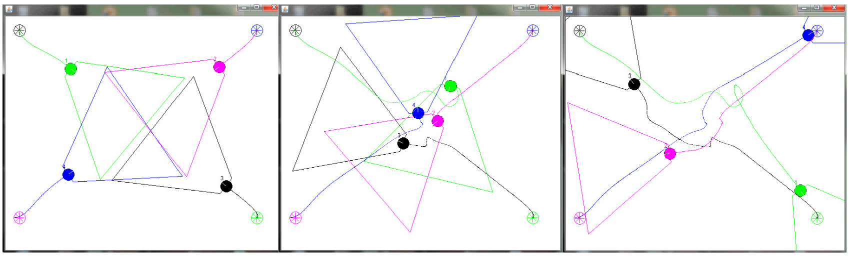


Fig. 2. Simulation 1: 4 robots without static obstacles

Figure 3 shows a similar situation but in this case there are four static obstacles, marked with black squares.

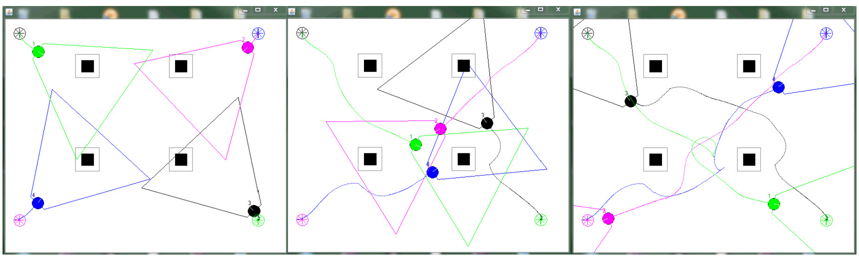


Fig. 3. Simulation 2: 4 robots with static obstacles

A video demonstration of practical experiment with Lego robots and two compiled versions of the platform that allow the simulation with robots can be obtained in http://idecona.ai2.upv.es/ (multimedia gallery and Resultados at Proyect menu)

3 Conclusions

A collision avoidance method that takes advantages and benefits of MAS has been presented in this work. This method is located one level above the traditional methods of obstacle avoidance where the management is performed locally and the possible communications between the local systems are solved functionally. The application of techniques provided by the area of artificial intelligence to the robotic area opens a wide range of possibilities that offers more natural results and gives human characteristics of communication like negotiation between robots. This work has succeeded in unifying concepts of agent theory with concepts from the area of mobile robotics, providing more intelligence to robots and offering solutions that otherwise can't be provided.

Acknowledgements. This work has been partially funded by the Ministerio de Ciencia e Innovación (Spain) under research projects DPI2011-28507-C02-01 and DPI2010-20814-C02-02.

References

1. Bernabeu, E.J.: Fast generation of multiple collision-free and linear trajectories in dynamic environments. IEEE Trans. Robotics 25(4), 967–975 (2009)
2. Bernabeu, E.J., Tornero, J., Tomizuka, M.: Collision prediction and avoidance amidst moving objects for trajectory planning applications. In: Proceedings of the IEEE Int. Conf. Robot. Automat., pp. 3801–3806 (2001)
3. Bruce, K.B., Cardelli, L., Pierce, B.C.: Comparing Object Encodings. In: Ito, T., Abadi, M. (eds.) TACS 1997. LNCS, vol. 1281, pp. 415–438. Springer, Heidelberg (1997)
4. Cameron, S., Culley, R.K.: Determining the minimum translational distance between two convex polyhedra. In: Proceeding of the IEEE Int. Conf. Robot. Automat., pp. 591–596 (1986)
5. Choi, Y.-K., Wang, W., Liu, Y., Kim, M.-S.: Continuous collision detection for two moving elliptic disks. IEEE Trans. Robotics 22(2), 213–224 (2006)
6. Java Agent Development Framework, `http://jade.tilab.com`
7. LEGO Mindstorms home page, `http://mindstorms.lego.com`
8. LeJOS: Java for LEGO Mindstorms, `http://lejos.sourceforge.net`
9. Russell, S., Norvig, P.: Artificial Intelligence: A modern approach (2003)
10. Schwarzer, F., Saha, M., Latombe, J.-C.: Adaptive dynamic collision checking for single and multiple articulated robots in complex environments. IEEE Trans. Robotics (2005)
11. SHARP Optoelectonic device GP2y0A21YK, `http://www.sharpsma.com/webfm_send/1208`
12. Tang, M., Kim, Y.J., Manocha, D.: C2A: Controlled conservative advancement for continuous collision detection of polygonal models. In: Proceedings of the IEEE Int. Conf. Robot. Automat., pp. 849–854 (2009)
13. Urmson, C., Anhalt, J., Bagnell, D., Baker, C.: Autonomous driving in urban environments: Boss and the urban challenge. Journal of Field Rob. 25(8), 425–466 (2008)
14. Van Leeuwen, J. (ed.): Computer Science Today. Recent Trends and Developments. LNCS, vol. 1000. Springer, Heidelberg (1995)

Parallel and Distributed Simulation of Large-Scale Cognitive Agents

Kashif Zia[1], Katayoun Farrahi[1], Alexei Sharpanskykh[2], Alois Ferscha[1], and Lev Muchnik[3,4]

[1] Pervasive Computing Institute, JKU Linz, Austra
[2] Department of Computer Science, VU University Amsterdam, Netherlands
[3] School of Business Administration, The Hebrew University of Jerusalem, Israel
[4] Beacon Tech Ltd.

Abstract. The large-scale simulation of complex cognitive agent based models simulated on realistic physical space require special computer processing considerations. In this paper, we explore the up-scaling of an agent based model to city-scale, by considering details about how to implement the models in a cellular automata (CA) in a parallel-distributed simulation (PDS) framework.

1 Distributed Agent Based Simulation

A large-scale agent based model (ABM) can be achieved by distributing the simulation world across a series of parallel processes. If the agents need explicit and tight integration with the underlaying space as that of an ABM in conjunction with a space represented by cellular automta (CA), the parallel and distributed simulation (PDS) using a shared memory multi-processor infrastructure provides a natural choice. CA provides a natural mechanism of space distribution across the processors in a regular grid style [1]. This can conveniently be achieved by using a high performance PDS enabled version of the ABM simulation tool – Repast for High Performance Computing (Repast HPC) [2]. Repast HPC provides built-in mechanisms of distributing a regular space into uniform processing units. The processes are then synchronized through shared memory using a message passing interface.

The successful up-scaling of an ABM to a PDS requires consideration about existing models in terms of agent resources. Contrary to the single processor case where all the resources *can* be accessed locally, PDS operates under modeler's provisioning of resources. In this paper, we present a mechanism of effective PDS in which our cognitive agent models, presented in [3], interact with each other and with the space to substantiate a spatially influenced behavior (density based mobility).

2 Cognitive Agents and Spatial Considerations

We present a modified version of the agent based models presented in [3], which have been scaled up for a city-scale simulation. The simulation is performed for

Y. Demazeau et al. (Eds.): PAAMS 2013, LNAI 7879, pp. 324–328, 2013.

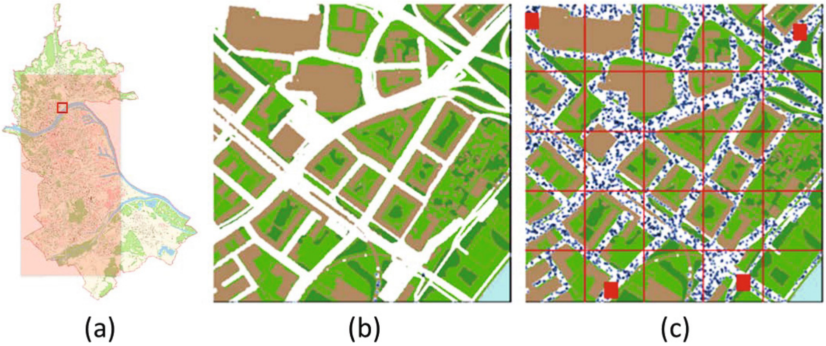


Fig. 1. Simulated area. (b) View of a neighborhood within the city (corresponding to the small red box in (a)). All white patches represent areas in which agents can move. (c) View of the physical space divided into 25 regions and a view with 5000 agents distributed uniformly over the space, with the points of attraction shown near the corners with red boxes.

a neighborhood in the city in Linz, Austria (see Fig. 1). The model components are described next; first we present the spatial considerations.

2.1 Space and Mobility

Each "region" corresponds to a space, visualized by a square in Figure 1 (c), and is simulated by a processor. A region consists of 100 x 100 cells, where cells can be thought of as pixels in an image. We define "points of attraction" (PoA), visualized by the red squares in Figure 1 (c), which "attract" agents towards them and guide the routing of agents. Each agent moves towards one PoA. The agent extracts this routing information from the underlying "cells" (spread from PoAs using floor field (FF) method [4] during simulation initialization). The FF provides the following information to the agents: *direction*, *distance* (hop count) and *route* (series of regions constituting the path towards a PoA). For mobility, an agent chooses a PoA which has minimum traveling time considering that the speed of an agent is dependent on density along its route.

2.2 Density Perception

Since the agent chooses a PoA depending on the accumulated route density (to choose the least dense route), the authenticity of regional density information is important. The difference between the actual value and the way it is "perceived" by an agent – quality of the information – decisively affect the decision making of an agent. An agent can perceive the density information in the following ways:

Personal Observation: The personal perception of an agent corresponds to its natural ability to observe its surroundings. Within a perceptible capability (e.g. visual and auditory range), an agent is considered to estimate the density

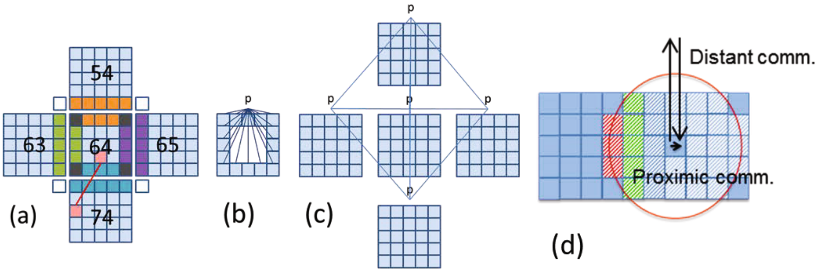


Fig. 2. Data Sharing across the Processes. (a) 4 out of 8 adjacent processes of process 64 with buffer size = 1. Cells in gray are being shared across more than one process. Adjacent processes also provide Network Synchronization (A2A in red). (b) Process level Parameters: can be collected and shared by all the resources of the process. (c) Inter-Process Parameters Sharing: Not supported by Repast-HPC; responsibility of all the processors to act on agents' behalf. (d) Proximic Communication of radius 3 (red colored circle of agent at the center). Enabled buffer synchronization across adjacent process (green), Disabled sync. across adjacent process (red). Also shows channels of inter-process distant communication.

around herself accurately. This personal observation acts as the *default* density perception of an agent of its current region and for the others in the absence of any "outside" information. The outside information is communicated through proximity-based and distant interaction.

Physical Proximity-Based Interaction: Within an interaction range of radius R, all agents can interact with each other and share information about their own perception.

Distant Interaction: Distant interaction has no spatial consideration, such as phone calls and messaging.

2.3 Route Selection

To reach a routing decision, an agent needs to (i) 'sense' its surrounding, (ii) interact with other agents in-proximity and at a distance. To execute a decision, it needs FF and neighborhood information. Next we present the implementation details for a CA based ABM to be executed in a shared-memory PDS.

3 PDS of Cognitive Agents

3.1 Spreading of FF and Local Mobility

As mobility should not be characterized by the availability of agents in proximity, we have used Moore's neighborhood [5] to realize city-scale pedestrian mobility. Repast HPC provides "buffer" sharing mechanism across adjacent processes, parametrized by the neighborhood type and the size of the buffer (see Fig. 2

(a)). During the FF spreading, originating from each PoA, a buffer size of "1" is sufficient. However, for agent's local mobility, a buffer size of "2" was required in our case. The local mobility desires collision avoidance which cannot be achieved without knowing the states of all the cells where an agent can jump to based on maximum speed it can acquire. The size of the cell determines the number of cells required to perform a movement step (with a speed). It varies from one simulation setting to the other. In addition of space and agents being buffered across adjacent processes, they can also be "networked" (see Fig. 2 (a)).

3.2 Sensing and Communication

For sensing (personal density perception) and communication (proximity-based and distant agent-to-agent (A2A) interaction), we need to look beyond the neighborhood. We start with "process-specific" data sharing – process-level data sharing (see Fig. 2 (b)) and inter-process data sharing (see Fig. 2 (c)). In this current model, we do not require process level parameters, as agent behavior is completely localized. However, the A2A distant communication (see Fig. 2 (d)) cannot be achieved without processors taking responsibility of sharing information with remote processors. This feature requires minimal usage however, due to the negative impact on simulation efficiency.

4 Discussion

Both sensing the surrounding and proximity-based interaction require specification of a radius (R). Within R, the agents would interact. In many cases, the radius would extend to adjacent processes. Here the restriction imposed by buffering size would be decisive (see Fig. 2 (d)). The following open issues are scenario dependent and the modeler needs to consider how to address them. What should be the value of R? How should region density be computed when agents are present on an adjacent process (but within R)? How should a large value of R be synchronized efficiently?

In most cases, the value of R should not be the same for sensing and agent interaction. The sensing is usually (human) perception-oriented. We have set it to 25 meters (25 cells) in all directions which may correspond to sense of vision in a street. If the buffer size is 2, most of the adjacent agents would not be counted as density of the current region – which is actually the case. For sensing, the larger the buffer size, the less the accuracy of perception. For proximity-based interaction, a buffer size of 2 would not work. Interaction is more than sensing as it requires some medium to share the information. If we consider a radius of 10 meters (10 cells) which roughly corresponds to vocal range, the synchronization turns out be computationally expensive. To resolve this trade-off, we settled with a buffer size of 5 cells for a 10 meter interaction radius. Although this is excessive for sensing, it works for interaction. The reason is that the spreading has a cascading effect often surpassing the speed of the agents. It means that the information diffuses in a transitive fashion even if the buffer size does not allow direct agent to agent communication.

References

1. Li, X., Zhang, X., Yeh, A., Liu, X.: Parallel cellular automata for large-scale urban simulation using load-balancing techniques. International Journal of Geographical Information Science 24, 803–820 (2010)
2. Collier, N.: Repast HPC Manual. Technical report, p.44 (November 2010)
3. Farrahi, K., Zia, K., Sharpanskykh, A., Ferscha, A., Muchnik, L.: Agent perception modeling for movement in crowds. In: Demazeau, Y., Ishida, T., Corchado, J.M., Bajo, J. (eds.) PAAMS 2013. LNCS, vol. 7879, pp. 73–84. Springer, Heidelberg (2013)
4. Kirchner, A., Schadschneider, A.: Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. Physica A: Statistical Mechanics and its Applications 312(1), 260–276 (2002)
5. Kretz, T., Schreckenberg, M.: Moore and more and symmetry. In: Pedestrian and Evacuation Dynamics 2005, pp. 297–308 (2005)

Author Index