LNAI 10978

Yves Demazeau · Bo An · Javier Bajo
Antonio Fernández-Caballero (Eds.)

Advances in Practical Applications of Agents, Multi-Agent Systems, and Complexity

The PAAMS Collection

16th International Conference, PAAMS 2018
Toledo, Spain, June 20–22, 2018
Proceedings

Springer

Lecture Notes in Artificial Intelligence 10978

Subseries of Lecture Notes in Computer Science

LNAI Series Editors

Randy Goebel
University of Alberta, Edmonton, Canada

Yuzuru Tanaka
Hokkaido University, Sapporo, Japan

Wolfgang Wahlster
DFKI and Saarland University, Saarbrücken, Germany

LNAI Founding Series Editor

Joerg Siekmann
DFKI and Saarland University, Saarbrücken, Germany

More information about this series at http://www.springer.com/series/1244

Yves Demazeau · Bo An
Javier Bajo · Antonio Fernández-Caballero (Eds.)

Advances in Practical Applications of Agents, Multi-Agent Systems, and Complexity

The PAAMS Collection

16th International Conference, PAAMS 2018
Toledo, Spain, June 20–22, 2018
Proceedings

Springer

Editors
Yves Demazeau
Centre National de la Recherche Scientifique
Grenoble
France

Javier Bajo
Universidad Politécnica de Madrid
Madrid
Spain

Bo An
Nanyang Technological University
Singapore
Singapore

Antonio Fernández-Caballero
Universidad de Castilla La Mancha
Albacete
Spain

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Artificial Intelligence
ISBN 978-3-319-94579-8 ISBN 978-3-319-94580-4 (eBook)
https://doi.org/10.1007/978-3-319-94580-4

Library of Congress Control Number: 2018948192

LNCS Sublibrary: SL7 – Artificial Intelligence

© Springer International Publishing AG, part of Springer Nature 2018
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by the registered company Springer International Publishing AG
part of Springer Nature
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Research on agents and multi-agent systems has matured during the past 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, became a necessity and was created almost two decades ago.

PAAMS, the International Conference on Practical Applications of Agents and Multi-Agent Systems, is the international yearly forum in which 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 2018 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: energy and security, engineering and tools, evaluation and ethics, negotiation and organizations, personalization and learning, simulation applications, simulation platforms, social networks and humans. Each paper submitted to PAAMS went through a stringent peer review by three members of the Program Committee composed of 121 internationally renowned researchers from 26 countries. From the 64 submissions received, ten were selected for full presentation at the conference; another ten papers were accepted as short presentations. In addition, a demonstration track featuring innovative and emergent applications of agent and multi-agent systems and technologies in real-world domains was organized. In all, 21 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, APPIA, NTU, and CNRS), and the Organizing Committee for their hard and highly valuable work. Their work contributed to the success of the PAAMS 2018 event. Thanks for your help – PAAMS 2018 would not exist without your contribution.

May 2018

Yves Demazeau
Javier Bajo
Bo An
Antonio Fernández-Caballero

Organization

General Co-chairs

Yves Demazeau	Centre National de la Recherche Scientifique, France
Bo An	Nanyang Technological University, Singapore
Javier Bajo	Polytechnic University of Madrid, Spain
Antonio Fernández-Caballero	Universidad de Castilla-La Mancha, Spain

Advisory Board

Paul Davidsson	Malmö University, Sweden
Keith Decker	University of Delaware, USA
Frank Dignum	Utrecht University, The Netherlands
Toru Ishida	University of Kyoto, Japan
Takayuki Ito	Nagoya Institute of Technology, Japan
Jörg P. Müller	Technische Universität Clausthal, Germany
Juan Pavón	Universidad Complutense de Madrid, Spain
Michal Pěchouček	Czech Technical University in Prague, Czech Republic
Franco Zambonelli	University of Modena and Reggio Emilia, Italy

Program Committee

Carole Adam	University of Grenoble, France
Emmanuel Adam	University of Valenciennes, France
Analia Amandi	University of Tandil, Argentina
Francesco Amigoni	Politecnico di Milano, Italy
Bo An (Co-chair)	Nanyang Technological University, Singapore
Luis Antunes	University of Lisbon, Portugal
Matteo Baldoni	University of Turin, Italy
Joao Balsa	University of Lisbon, Portugal
Cristina Baroglio	University of Turin, Italy
Nick Bassiliades	University of Thessaloniki, Greece
Jeremy Baxter	QinetQ, UK
Michael Berger	DocuWare AG, Germany
Olivier Boissier	Ecole des Mines des Saint Etienne, France
Rafael Bordini	Pontifical University of Rio Grande do Sul, Brazil
Vicente Botti	Polytechnic University of Valencia, Spain
Bruno Bouchard	University of Québec at Chicoutimi, Canada
Lars Braubach	Universität Hamburg, Germany
Sven Brueckner	Axon AI, USA
Javier Carbó	Carlos III University of Madrid, Spain

Luis Castillo	University of Caldas, Colombia
Georgios Chalkiadakis	Technical University of Crete, Greece
Carlos Chesnevar	Universidade Nacional des Sur, Argentina
Rafael Corchuelo	University of Seville, Spain
Paul Davidsson	Malmö University, Sweden
Keith Decker	University of Delaware, USA
Yves Demazeau (Co-chair)	Centre National de la Recherche Scientifique, France
Frank Dignum	University of Utrecht, The Netherlands
Jürgen Dix	Clausthal University of Technology, Germany
Alexis Drogoul	Institut de Recherche pour le Développement, Vietnam
Julie Dugdale	University of Grenoble, France
Johannes Fähndrich	Technical University of Berlin, Germany
Rubven Fuentes	Complutense University of Madrid, Spain
Katsuhide Fujita	Tokyo University of Agriculture and Technology, Japan
Naoki Fukuta	Shizuoka University, Japan
Sylvain Giroux	University of Sherbrooke, Canada
Marie-Pierre Gleizes	University of Toulouse, France
Daniela Godoy	University of Tandil, Argentina
Jorge J. Gómez-Sanz	Complutense University of Madrid, Spain
Vladimir Gorodetski	University of Saint Petersbourg, Russia
Charles Gouin-Vallerand	Télé-Université du Québec, Canada
Vincent Hilaire	University of Belfort-Montbeliard, France
Koen Hindrinks	University of Delft, The Netherlands
Martin Hofmann	Lockheed Martin, USA
Tom Holvoet	Catholic University of Leuven, Belgium
Jomi Hubner	Universidad Federale de Santa Catarina, Brazil
Piot Jedrzejowicz	Gdynia Maritime University, Poland
Yichuan Jiang	Southeast University of Nanjing, China
Xiolong Jin	Chinese Academy of Science, China
Vicente Julian	Polytechnic University of Valencia, Spain
Ryo Kanamori	Nagoya University, Japan
Takahiro Kawamura	Toshiba, Japan
Chris Kiekintveld	University of Texas at El Paso, USA
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
Robin Lamarche-Perrin	University of Paris 6, France
Paulo Leitao	Polytechnic Institute of Bragança, Portugal
Brian Logan	University of Nottingham, UK
Henrique Lopes Cardoso	University of Porto, Portugal
Miguel Angel Lopez-Carmona	University of Alcala, Spain

Maite Lopez-Sanchez	University of Barcelona, Spain
Rene Mandiau	University of Valenciennes, France
Wenji Mao	Chinese Academy of Science, China
Philippe Mathieu	University of Lille, France
Eric Matson	Purdue University, USA
Ceti Mericli	Carnegie Mellon University, USA
Fabien Michel	University of Reims, France
José M. Molina	Carlos III University of Madrid, Spain
Mirko Morandini	University of Trento, Italy
Bernard Moulin	Laval University, Canada
Jean-Pierre Muller	CIRAD, France
Joerg Mueller	Clausthal University of Technology, Germany
Robert Neches	ISI, IARPA, USA
Ngoc Thahn Nguyen	Wroclaw University of Technology, Poland
Paolo Novais	University of Minho, Portugal
Akihiko Ohsuga	University of Electro-Communications, Japan
Eugenio Oliveira	University of Porto, Portugal
Andrea Omicini	University of Bologna, Italy
Mehmet Orgun	Macquarie University, Australia
Sascha Ossowski	Rey Juan Carlos University, Spain
Van Parunak	ABC Research, USA
Juan Pavon	Complutense University of Madrid, Spain
Terry Payne	University of Liverpool, UK
Sébastien Picault	University of Lille, France
Faruk Polat	Middle East Technical University, Turkey
Luis Paulo Reis	University of Porto, Portugal
Alessandro Ricci	University of Bologna, Italy
Deborah Richards	Macquarie University, Australia
Ana Paula Rocha	University of Porto, Portugal
Juan Rodriguez Aguilar	Artificial Intelligence Research Institute, Spain
Sebastian Rodriguez	Universidad Tecnologica Nacional, Argentina
Silvia Schiaffino	University of Tandil, Argentina
Franciszek Seredynski	Cardinal Stefan Wyszynski University, Poland
Jaime Sichman	University of Sao Paulo, Brazil
Petr Skobelev	Smart Solutions, Russia
Leandro Soriano Marcolino	University of Southern California, USA
Sonia Suárez	University of La Coruna, Spain
Jose Such	University of Lancaster, UK
Toshiharu Sugawara	Waseda University, Japan
Matthew Taylor	Washington State University, USA
Simon Thomson	British Telecom, UK
Ingo Timm	University of Trier, Germany
Viviane Torres da Silva	Universidad Federal Fluminente, Brazil
Paolo Torroni	University of Bologna, Italy
Ali Emre Turgut	Middle East Technical University, Turkey
Domenico Ursino	University of Reggio Calabria, Italy

Laszlo Varga	Computer and Automation Research Institute, Hungary
Wamberto Vasconselos	University of Aberdeen, UK
Laurent Vercouter	University of Rouen, France
Jacques Verriet	TNO, The Netherlands
José Villar	University of Oviedo, Spain
Wayne Wobcke	University of New South Wales, Australia
Gaku Yamamoto	IBM, Japan
Neil Yorke-Smith	American University of Beirut, Lebanon
Franco Zambonelli	University of Modena, Italy
Laura Zavala	University of Maryland, USA
Jinyu Zhang	University of Nanjing, China

Organizing Committee

Javier Bajo (Co-chair)	Universidad Politécnica de Madrid, Spain
Antonio Fernández-Caballero (Co-chair)	Universidad de Castilla-La Mancha, Spain
Pascual González	Universidad de Castilla-La Mancha, Spain
Elena Navarro	Universidad de Castilla-La Mancha, Spain

Local Organizing Committee

José Carlos Castillo	Universidad Carlos III de Madrid, Spain
Beatriz García-Martínez	Universidad de Castilla-La Mancha, Spain
María Teresa López	Universidad de Castilla-La Mancha, Spain
Víctor López-Jaquero	Universidad de Castilla-La Mancha, Spain
Arturo Martínez-Rodrigo	Universidad de Castilla-La Mancha, Spain
José Pascual Molina	Universidad de Castilla-La Mancha, Spain
Francisco Montero	Universidad de Castilla-La Mancha, Spain
Encarnación Moyano	Universidad de Castilla-La Mancha, Spain
Miguel Oliver	Universidad de Castilla-La Mancha, Spain
José Manuel Pastor	Universidad de Castilla-La Mancha, Spain
Miguel Ángel Teruel	Universidad de Castilla-La Mancha, Spain
Francisco José Vigo Bustos	Universidad de Castilla-La Mancha, Spain

PAAMS 2018 Sponsors

Contents

Demo Papers

Invited Speaker

Hammer or Tongs: How Best to Build Agent-Based Models?

Michael J. North

1147 Greensfield Drive, Naperville, IL 60563, USA

Abstract. Agent-based modeling has been widely applied to solve problems in contexts ranging from exploratory research studies to focused industrial practice. As the application of agent-based modeling has grown, developers have advocated a wide and diverse variety of development techniques. Some of these techniques inform elements of model creation, validation, or use. Others are comprehensive development methodologies. The popularity of agent-based modeling makes it impossible to include every development technique mentioned in the literature. Instead, this paper focuses on providing a representative sample that includes the techniques with the most practical utility along with those that are widely used or have received substantial attention. The sample reviewed in this paper includes techniques advocated specifically for agent-based modeling along with general software development approaches. Due to their importance, verification and validation techniques for agent-based modeling are specially highlighted. The paper concludes with a summary and recommendations on how best to build agent-based models.

Keywords: Agent-based modeling · Agent-based simulation
Software development · Development methodologies

1 Introduction

1.1 Remit

Agent-based modeling has been widely adopted as a modeling technique (Borshchev and Filippov 2004; North and Macal 2007; Kravari and Bassiliades 2015; Abar et al. 2017). As the application of agent-based modeling has grown, developers have advocated a wide and diverse variety of development techniques. Some of these techniques inform elements of model creation, validation, or use (Bert et al. 2014; Giancarlo et al. 2005). Others are comprehensive development methodologies (Wooldridge 1997; Santos et al. 2017). In this paper I survey the range of such techniques and offer recommendations as to which to choose for specific projects.

The popularity of agent-based modeling makes it impossible to include every development technique mentioned in the literature. Instead, this paper focuses on providing a representative sample that includes the techniques with the most practical

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 3–11, 2018.
https://doi.org/10.1007/978-3-319-94580-4_1

utility along with those that are widely used or have received substantial attention. The sample in this paper includes techniques advocated specifically for agent-based modeling along with general software development approaches[1].

2 Survey

2.1 General Methods for Developing Software

Participatory agent-based modeling (North and Macal 2007) notwithstanding, agent-based models are usually implemented as software. The sophistication of this software can range from intentionally simple (Olson and Horn 2011; Gkaras and Dimitrios 2017) to quite complicated (Collier and North 2012; Kiran 2014).

Regardless of the level of complication, the common software development challenges Brooks observed as early as 1975 apply including the need for pilot systems or prototypes; the need for incremental development; the tendency for process or code fixes to cause further breakage[2]; and the lack of a single "silver bullet" process or technology that can tame the inherent wildness of software development (Brooks 1975). Furthermore, Brooks' recent observation is worth noting (Brooks quoted in Kelly 2010):

> Software is not the exception; hardware is the exception. No technology in history has had the kind of rapid cost/performance gains that computer hardware has enjoyed. Progress in software is more like progress in automobiles or airplanes: We see steady gains, but they're incremental.

In widespread practice, these incremental gains have been significantly driven by a shift from structured waterfall[3] development as codified by organizations such as the Project Management Institute (2008) to agile development techniques (Beck et al. 2001) per the Agile Manifesto:

- Individuals and interactions over processes and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.

More recent threads along these lines include software craftspersonship (Software Craftsmanship Manifesto Signatories 2009), which focuses development processes on the act of writing code and emphasizes supporting the community of code writers, and

[1] This paper focuses on agent-based modeling. References to the related multi-agent systems technique are limited since this field has different, although allied, goals (Nikolic and Ghorbani 2011; Fortino and North 2013).

[2] The anonymous (2000) refrain, "99 little bugs in the code; 99 little bugs; take one down, patch it around; 127 little bugs in the code," is memorable in this regard.

[3] Waterfall development is a family of classic structured, multistage, software development methodologies that emphasize minimizing the amount of rework on the theory that costs rise exponentially as a mistake in a given stage propagates to later stages (Project Management Institute 2008). Agile development methodologies reject this theory (Beck et al. 2001; Software Craftsmanship Manifesto Signatories 2009; Collier and Ozik 2013) and instead focus on rapid refocusing based on feedback from real code.

test-driven development (Collier and Ozik 2013), which focuses on test writing as the main factor guiding software development.

These approaches vary substantially in their recommendations on the day-to-day practice of software development. Nonetheless, all of these approaches embrace iterative development; encourage maximal feedback as soon as practical; demand the regular use of prototyping; and emphasize incremental completion of rigorous verification and validation (V&V) as soon as possible in the development process.

In addition to the methods described above, some gains have also come from the reemergence of interest in applying formal methods for automating proofs to software development (Edwards 2016), such as the application of the calculus of inductive constructions to programming (Chlipala 2017). This family of approaches uses theoretical mathematical models to prove various properties of software designs or implementations. While there are mathematical frameworks that might be used as a foundation for formal agent-based model development (North 2014), these approaches have not taken hold in the larger agent-based modeling community of practice and will not be addressed further here.

2.2 General ABM Development Observations

Many papers have provided general observations on agent-based model development. For example, in addition to advocating for agent-based modeling as a simulation technique, Bonabeau (2002) encourages the development of use-case-specific models rather than multi-focus simulations. They also emphasize the importance of designing with the proper amount of abstraction. A 2009 *Nature* editorial recommended nested modular architectures and standard test cases. Bersini (2012) details how to apply the widely used Unified Modeling Language (UML) to agent-based modeling. North (2014) offers a formal framework for analyzing, and in many cases proving, the time and space performance of agent-based models.

Rovere et al. (2016) provide a variety of recommendations using an agricultural agent-based model as an example. They detail the benefits they received from the use of design patterns; discuss how they substantially reduced their model execution time; review their model initialization procedures; cover the source code quantification tools they used; and relate their V&V methods. Their overall recommendations are to take care to choose a relevant agent-based modeling framework, document the modeling effort using industrial-grade tools, use software development best practices such as design patterns, generate application scenarios using repeatable automated scripts, and integrate V&V into every development activity rather than saving it for the end.

2.3 Methods for Developing Agent-Based Models

Table 1 lists a range of published methods for developing agent-based models.

Table 1. Published agent-based model development methods.

Citation	Type[a]	Summary
Wooldridge (1997)	General structured	The author describes a waterfall development methodology based on modal logic
Jennings (2000) and Jennings (2001)	General structured	The author describes a waterfall development methodology based on agent-oriented decompositions. The methodology is intended for general software, not just models. However, it is directly applicable to agent-based modeling
Jo (2001)	General structured	The author describes a waterfall development methodology based on the Belief-Desire-Intention architecture
Grimm et al. (2005)[b]	Domain-specific to ecology	The authors offer a structured method based on a rigorous standardized description format and ecology-related design patterns. Their standardized model description format has been widely adopted in ecology and beyond
Yilmaz and Ören (2007)	General structured	The authors applied systems engineering to interactive agent-based modeling
Siegfried et al. (2009)	General structured	The authors introduce a formal notation to describe basic agent-based models and show how it can be used to design and develop such models
Wooldridge et al. (2010)	General structured	The authors describe an intensive waterfall development methodology based on “interacting roles.”
Niazi (2011)	General agile	The author details a comprehensive development methodology that is patterned after the scientific method
Nikolic and Ghorbani (2011)	General structured	The authors apply systems analysis to the development of models of large-scale socio-technical systems such as supply chains
Siebers et al. (2013)	General structured	The authors of this presentation apply standard software engineering to the development of agent-based models
Ghorbani, et al. (2013)	Domain-specific to economics and social systems	The authors apply the institutional analysis and development framework from economics to yield a detailed ontology and process for agent-based modeling
North and Macal (2013)	General agile	The authors offer a highly iterative approach to agent-based modeling based on agile software development methods
Juarez (2014)	Domain-specific to developmental economics	The authors use theories from developmental economics to produce a detailed ontology for agent-based developmental economics modeling
Lange et al. (2017)	Domain-specific to urban studies	The authors of this presentation use concepts from urban studies to produce an ontology for agent-based urban systems modeling.
Santos et al. (2017)	Domain-specific to traffic studies	The authors apply theories from roadway traffic flow studies to create an ontology and model-driven engineering process for agent-based traffic modeling.

[a]Methodology type is domain-specific, general structured, or general agile. If the methodology is domain-specific then the primary domain is named.

[b]Grimm et al. (2005) also describe a “Medawar zone” by analogy with the concept of the same name from the analysis of the returns from scientific studies. The Medawar zone is an area of medium model complexity that is said to often have the highest payoff. Models simpler than those in the Medawar zone do not have enough detail to yield a high payoff. Models more complex than those in the Medawar zone are too complex to easily learn from or use, thus again limiting the payoff.

2.4 ABM Verification and Validation

V&V is an indispensible part of any practical agent-based modeling project[4]. Table 2 lists a range of published methods for agent-based model V&V.

Table 2. Published agent-based model V&V methods.

Citations	Type[a]	Summary
Giancarlo et al. (2005)	General structured	The authors detail a discrete-event software framework for runtime validation of agent-based model behavior
Xiang et al. (2005)	Domain-specific to physical science models	The authors relate a conceptual framework for applying traditional V&V concepts to agent-based models of physical science problems
Klügl (2008)	General structured	The author discusses how to use face validation and a set of standard statistical techniques to perform V&V on agent-based models
Niazi et al. (2009)	General agile	The authors describe a detailed framework for V&V iterative development of agent-based models. This is the V&V component of Niazi (2011)
Sargent (2010)	General structured	The author provides an outstanding synthesis of traditional simulation V&V which is directly applicable to agent-based models
Denz (2013)	General structured	The author introduces a method to extract behavioral descriptions from instrumented agent-based models using process mining. The author then shows how to use the results to validate the behavior of the model undergoing V&V
Bert et al. (2014)	General agile	The authors present a series of lessons learned from validating an agent-based model. The recommendations are summarized in an earlier section

[a]As before, methodology type is domain-specific, general structured, or general agile. If the methodology is domain-specific then the primary domain is named. Here general structured methodologies assume that agent-based models are completed artifacts that need to undergo various forms of Popperian falsification.

2.5 ABM Toolkit Development

In addition to the large and growing number of agent-based modeling projects, there are a substantial number of projects focused on developing software for agent-based modeling (Kravari and Bassiliades 2015; Abar et al. 2017). North et al. (2006) provide a survey of the literature on lesions learned from software tools, frameworks, and

[4] It *should* be an indispensible part of any theoretical agent-based modeling project as well!.

environments for agent-based modeling. Their conclusions are that fine grained object-orientation is highly desirable for agent-based modeling, using a cross-platform language is helpful, allowing users to select their own architecture is important, support for design-patterns is useful, and offering access to an extensive list of third party libraries is also valuable.

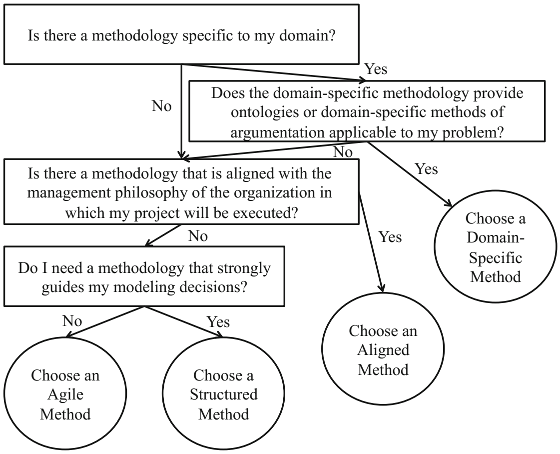


Fig. 1. An agent-based modeling methodology decision tree.

3 Conclusions

3.1 "Survey Says"[5]

The techniques surveyed in this paper can be broken up into domain-specific methods and domain-independent methods. The former can save time and energy by offering already-completed work. The latter can provide flexibility for genuinely custom projects. In either case, V&V is an essential component of all agent-based modeling.

The domain-specific methods take advantage of whatever natural structure exists in their area of specialization to accelerate and channel the agent-based modeling process. The acceleration can include the partial or even full ontologies to provide ready-made scaffolding for modeling. The channeling can include specialized V&V methods based on discipline-specific standards of evidence or domain-specific methods of argumentation. Either way, substantial V&V is required in all cases.

The domain-independent methods can be further subdivided into structured and iterative methods. The structured methods ultimately inherit from traditional project management methods (Project Management Institute 2008). In the best of circumstances, they can provide a checklist-driven development process that may be appealing to new agent-based modelers or those that need help narrowing down the

[5] This quote is a tagline from the long-running American television program *Family Feud*.

virtually unlimited flexibility of agent-based modeling. The iterative methods are generally agile methodologies (Beck et al. 2001) specialized for agent-based modeling. Most of the latter approaches embrace incremental development, encourage receiving as much feedback as possible as soon as practical, and recommend regular use of prototyping. Both structured and iterative methods require rigorous V&V.

3.2 How to Choose

The unbounded number of potential agent-based modeling projects multiplied times the wide range of development approaches yields an effectively infinite number of possible methodology decision criteria. Narrowing this down, the practical factors to consider include saving development time by choosing a methodology with immediately usable components and getting help with the many choices inherent in agent-based modeling by selecting a methodology that structures modeling decisions. Methods that focus on Grimm et al. (2005)'s Medawar zone are also recommended.

Figure 1 shows a decision tree for selecting an agent-based modeling methodology. This decision tree can be used independently for selection of both an overall development methodology and a V&V methodology, should the chosen overall development methodology not include sufficient V&V. Once a recommendation node (i.e., circle) is selected in Fig. 1 then a method of the appropriate type can be selected from either Table 1 or Table 2, as needed.

References

Abar, S., Theodoropoulos, G.K., Lemarinier, P., O'Hare, G.M.P.: Agent based modelling and simulation tools: a review of the state-of-art software. Comput. Sci. Rev. **24**, 13–33 (2017)

Anonymous: 99 Little Bugs in the Code (2000). http://www.gdargaud.net/Humor/QuotesProgramming.html

Beck et al.: Manifesto for Agile Software Development (2001). http://agilemanifesto.org/. Accessed on 20 Mar 2018

Bersini, H.: UML for ABM. J. Artif. Soc. Soc. Simul. **15**(1), 9 (2012)

Bert, F.E., Rovere, S.L., Macal, C.M., North, M.J., Podestá, G.P.: Lessons from a comprehensive validation of an agent-based model: the experience of the pampas model of argentinean agricultural systems. Ecol. Model. **274**, 284–298 (2014)

Bonabeau, E.: Agent-based modeling: methods and techniques for simulating human systems. Proc. Nat. Acad. Sci. U.S.A. **99**(Suppl. 3), 7280–7287 (2002)

Borshchev, A., Filippov, A.: From system dynamics and discrete event to practical agent based modeling: reasons, techniques, tools. In: 22nd International Conference of the System Dynamics Society, Oxford, 25–29 July 2004

Brooks, F.: The Mythical Man-Month. Addison-Wesley, Boston (1975)

Chlipala, A.: Certified Programming with Dependent Types. MIT Press, Cambridge (2017)

Collier, N.T., Ozik, J.: Test-driven agent-based simulation development. In: Proceedings of the 2013 Winter Simulation Conference, 8–11 December 2013. IEEE Press, Washington, D.C. (2013)

Collier, N.T., North, M.J.: Parallel agent-based programming with repast for high performance computing. Simulation **89**, 1215–1235 (2012)

Denz, N.: Process-oriented analysis and validation of multi-agent-based simulations: concepts and case studies. Dissertation, University of Hamburg (2013)

Edwards, C.: Automating proofs. Commun. ACM **59**(4), 13–15 (2016)

Fortino, G., North, M.J.: Simulation-based development and validation of multi-agent systems: AOSE and ABMS approaches. J. Simul. **7**(3), 137–143 (2013)

Ghorbani, A., Bots, P., Dignum, V., Dijkema, G.: MAIA: a framework for developing agent-based social simulations. J. Artif. Soc. Soc. Simul. **16**(2), 9 (2013)

Giancarlo, F., Garro, A., Russo, W.: A Discrete-event simulation framework for the validation of agent-based and multi-agent systems. In: dagli Oggetti agli Agenti Simulazione e Analisi Formale di Sistemi Complessi, Camerino, 14–16 November 2005, pp. 75–84 (2005)

Gkaras, G., Dimitrios, Y.: Teaching 'emergence' with netlogo in a high school project about complex systems and climate change. In: Proceedings of the 3rd International Conference for the Promotion of Educational Innovation, Larissa, October 2017

Grimm, V., Revilla, E., Berger, U., Jeltsch, F., Mooij, W.M., Railsback, S.F., Thulke, H.-H., Weiner, J., Wiegand, T., DeAngelis, D.L.: Pattern-oriented modeling of agent-based complex systems: lessons from ecology. Science **310**(5750), 987–991 (2005)

Jennings, N.R.: On agent-based software engineering. Artif. Intell. **117**, 277–296 (2000)

Jennings, N.R.: An agent-based approach for building complex software systems. Commun. ACM **44**(4), 35–41 (2001)

Jo, C.-H.: A seamless approach to the agent development. In: Proceedings of the 2001 ACM Symposium on Applied Computing, Las Vegas, pp. 641–647 (2001)

Juarez, F.W.C.: Agent-based modeling techniques for development economics. SSRN Electro. J., 1–16 (2014)

Kelly, K.: Master planner: fred Brooks shows how to design anything. Wired, July 2010. https://www.wired.com/2010/07/ff-fred-brooks/

Kiran, M.: Using FLAME toolkit for agent-based simulation: case study sugarscape model. Computing Research Repository https://arxiv.org/abs/1408.3441 (2014)

Klügl, F.: A validation methodology for agent-based simulations. In: Proceedings of the 2008 ACM Symposium on Applied Computing, Fortaleza, 16–20 March 2008, pp. 39–43 (2008)

Kravari, K., Bassiliades, N.: A survey of agent platforms. J. Artif. Soc. Soc. Simul. **18**(1), 11 (2015)

Lange, K., Oskam, I., Korevaar, G., Herder P.: Towards a design research methodology: agent-based model development through stakeholder participation in the case of decentralized urban vermicomposting facilities. In: 2017 International Symposium on Sustainable Systems and Technology, Chicago, 29 June 2017. http://programme.exordo.com/isie2017/delegates/presentation/382/. Accessed 17 Mar 2018

Nature Editorial: A model approach. Nature **460**, 667 (2009)

Niazi, M.: Towards a novel unified framework for developing formal, network and validated agent-based simulation models of complex adaptive systems. Dissertation, University of Stirling (2011)

Niazi, M., Hussain, A., Kolberg, M.: Verification & validation of agent based simulations using the VOMAS (virtual overlay multi-agent system) approach. In: Proceedings of the Third Workshop on Multi-Agent Systems and Simulation, Torino, 7–11 September 2009

Nikolic, I., Ghorbani, A.: A method for developing agent-based models of socio-technical systems. In: 2011 IEEE International Conference on Networking, Sensing and Control, Delft, 11–13 April 2011

North, M.J.: A theoretical formalism for analyzing agent-based models. Complex Adapt. Sys. Model. **2**(1), 3 (2014)

North, M.J., Collier, N.T., Vos, R.J.: Experiences creating three implementations of the repast agent modeling toolkit. ACM Trans. Model. Comput. Simul. **16**(1), 1–25 (2006)

North, M.J., Macal, C.M.: Managing Business Complexity: Discovering Strategic Solutions with Agent-Based Modeling and Simulation. Oxford University Press, New York (2007)
North, M.J., Macal, C.M.: Product and process patterns for agent-based modeling and simulation. J. Simul. **8**, 25–36 (2013)
Olson, I.C., Horn, M.S.: Modeling on the table: agent-based modeling in elementary school with NetTango. In: Proceedings of the 10th International Conference on Interaction Design and Children, Ann Arbor, 20–23 June 2011
Project Management Institute: Project Management Body of Knowledge, 4th edn. Project Management Institute, Newtown Square (2008)
Rovere, S.L., North, M.J., Podestá, G.P., Bert, F.E.: Practical points for the software development of an agent-based model of a coupled human-natural system. IEEE Access **4**, 4282–4298 (2016)
Santos, F., Nunes, I., Bazzan, A.L.C.: Model-driven engineering in agent-based modeling and simulation: a case study in the traffic signal control domain (extended abstract). In: Das, S., Durfee, E., Larson, K., Winikoff, M. (eds.) Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems, Sao Paulo, 8–12 May 2017, pp. 1725–1727 (2017)
Sargent, R.G.: Verification and validation of simulation models. In: Johansson, B., Jain, S., Montoya-Torres, J., Hugan, J., Yücesan, E. (eds.) Proceedings of the 2010 Winter Simulation Conference, Baltimore, 5–8 December 2010, pp. 166–183 (2010)
Siebers, P.O., Skatova, A., Vu, T., Aickelin, U.: A software engineering approach to designing agent-based models. University of Nottingham Presentation, 9 September 2013. http://www.cs.nott.ac.uk/~pszps/bemoossig/slides/BEMooSSIG-Siebers2013-09-13v3.pdf. Accessed 17 Mar 2018
Siegfried, R., Lehmann, A., Khayari, R.E.A., Kiesling, T.: A reference model for agent-based modeling and simulation. In: Proceedings of the 2009 Spring Simulation Multiconference, San Diego, 22–27 March 2009
Software Craftsmanship Manifesto Signatories: Manifesto for software craftsmanship (2009). http://manifesto.softwarecraftsmanship.org/. Accessed 20 Mar 2018
Wooldridge, M.: Agent-based software engineering. IEE Proc. – Softw. **144**(1), 26–37 (1997)
Wooldridge, M., Jennings, N.R., Kinny, D.: The Gaia methodology for agent-oriented analysis and design. Auton. Agent. Multi-Agent Syst. **3**(3), 285–312 (2010)
Xiang, X., Kennedy, R., Madey, G., Cabaniss S.: Verification and validation of agent-based scientific simulation models. In: Agent-Directed Simulation Conference, San Diego, April 2005, pp. 27–34 (2005)
Yilmaz, L., Ören, T.I.: Agent-directed simulation systems engineering. In: Proceedings of the 2007 Summer Computer Simulation Conference, San Diego, 16–19 July 2007, pp. 897–904 (2007)

Check for updates

Towards Autonomous AI Systems for Resource Management: Applications in Industry and Lessons Learned

Petr Skobelev[1,2,3](✉)

[1] Smart Solutions, Ltd. - Knowledge Genesis Group, Samara, Russia
skobelev@kg.ru
[2] Institute for the Control of Complex Systems of Russian Academy of Sciences, Samara, Russia
[3] Samara State Technical University, Samara, Russia

Abstract. Complexity of modern resource management is analyzed and related with a number of decision makers, high variety of individual criteria, preferences and constraints, interdependency of all operations, etc. The overview of existing methods and tools of Enterprise Resource Planning is given and key requirements for resource management are specified. The concept of autonomous Artificial Intelligence (AI) systems for adaptive resource management based on multi-agent technology is discussed. Multi-agent model of virtual market and method for solving conflicts and finding consensus for adaptive resource management are presented. Functionality and architecture of autonomous AI systems for adaptive resource management and the approach for measuring adaptive intelligence and autonomy level in these systems are considered. Results of delivery of autonomous AI solutions for managing trucks and factories, mobile teams, supply chains, aerospace and railways are presented. Considerable increase of enterprise resources efficiency is shown. Lessons learned from industry applications are formulated and future developments of AI for solving extremely complex problems of adaptive resource management are outlined.

Keywords: Complexity · Artificial Intelligence · Autonomous systems Adaptability · Resource management · Multi-agent technology Demand-Resource Networks · Self-organization · Competitive equilibriums Real-time economy

1 Introduction

The growing complexity of the modern economy is already widely recognized and associated with the increased uncertainty and dynamics of demand and supply, where disruptive events become rather norm than exception [1].

However, until now typical reactions to unpredictable events in business have been widely the following: bringing oversized resources, increasing costly safety stocks or injecting slack-time. The result of such reactions is decreased service quality, reduced

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 12–25, 2018.
https://doi.org/10.1007/978-3-319-94580-4_2

utilization of resources, lost orders or increased costs pushed onto the customer and, finally, drastic decrease in business efficiency and competitiveness [2].

This challenge demands new type of industrial solutions for resource management that need to support a high level of adaptability and responsiveness [3].

Nowadays it becomes possible to address the challenge with the use of autonomous Artificial Intelligence (AI) systems working continuously and able to make decisions on their own - providing resource allocation, scheduling, optimization, co-ordination and control of results in real time. AI technologies are becoming the hot topic in science and technology but many R&D projects on autonomous systems are still mainly focused on robots and unmanned vehicles [4]. Some other areas of AI technologies include big data analytics, pattern recognition and machine vision, language processing, machine learning, etc. However, AI technologies for resource management are usually not included in this list – probably, due to the lack of industrial applications.

In this paper we present practical results of solving complex problems of adaptive resource management in industry with the use of autonomous AI systems based on multi-agent technology. These systems can be categorized, from one hand, as "weak (narrow)" AI systems that augment intelligence of humans in business. But, from the other hand, they exhibit more generic *emergent intelligence* based on self-organization of agents. This new property of developed systems can be considered as the first step to "strong" AI which is regulated by human-like cognitive abilities including the ability to modify its own functioning by learning from experience and reprogramming the AI to achieve better results. These systems will be used in future for managing not only humans but also swarms of unmanned vehicles of any kind: taxi, trucks, sea vessels, drones, satellites, etc. (https://www.youtube.com/watch?v=r7vKK9XnTCE).

In Sects. 2, 3 of the paper we analyze complexity of modern resource management and show the limitations of existing models, methods and tools for supporting the growing complexity and adaptability of business. In Sect. 4 we consider the concept of autonomous AI system for adaptive resource management based on the virtual market of agents and the method of solving conflicts by multi-iterative auctions. The functionality and architecture of autonomous AI solutions for adaptive resource management are presented in Sect. 5. The examples of developed autonomous AI solutions for managing trucks and factories, mobile teams, supply chains, aerospace and railways are presented in Sect. 6, showcasing increase of enterprise efficiency of up to 15–40%. The ways of measuring adaptability and level of autonomy of AI solutions, business experience and lessons learned are presented in Sect. 7.

Conclusion summarizes the benefits of developed solutions and outlines the plan of future developments.

2 Complexity of Modern Resource Management

Many modern enterprises are now looking for the better quality and efficiency of their products and services, transforming their businesses into new models: demand-oriented; subject to intense change and volatility; distributed, usually globally; operating in real time; needing agility to re-plan and re-allocate scarce resources; seeking improvement; concerned about costs and enterprise security, etc.

To identify the key factors that significantly increase complexity in resource management, let us consider the real cases solved in the past [5]:

- **Managing Trucks for a National Logistics Company:** 500 trucks on the road, 100 orders daily, horizon up to 10 days, 600 destinations, criteria: profit, service quality, preferred loading/unloading time (fixed and variable), backhauls, consolidation, vehicle capacity, loading sequence (backdoor and curtain sider), constraint stressing, planning in continuous mode, handling driver breaks, cross-docking, trucks utilization, drivers regulations, load compatibility.
- **Supply Chain of a Food Delivery Company:** up to 10 thousand products, 5 hundred nodes in the network, including 5 factories, 300 storages, 1000 shops, 500 delivery channels, 2 thousand orders a day, horizon up to 10 days; criteria: service level, minimum costs and penalties for manufacturing, transport and storages.
- **Field Services of Technicians for a Regional Gas Company:** 50–250 orders a day, 43 teams for region equipped with special equipment; scheduling horizon – 5 days, team working hours - 8–12 h; time for reaction on event – up to 1 min, criteria: the number of executed orders, minimum of empty miles, balance of resource loading.
- **Delivering Cargo for the International Space Station:** 3500 types of cargos (fuel, air, food, equipment, etc.), about 4–8 spaceships yearly, time of mission, number of astronauts, orbits and ballistics, sun activity, norms of cargo use, period of 6–12 months, criteria: safety and security of cosmonauts, reliability of equipment, timetable for scientific programs, capacity of ships and fuel tanks, number of ports.
- **Workshop of a Jet Engine Manufacturing Factory:** 150 workers and 300 units of equipment, one component requires from 10 to 25 mechanical, thermal and other operations, 100 tasks per day, one plan for a worker for 2–3 days includes 100 operations, horizon up to 6 months. Considering specifics of orders, sequences of operations, skills of workers, availability of equipment, materials, etc. Criteria: delivery in time and in budget, quality of products, no broken parts and products.
- **Project Management at an Airspace Enterprise:** one department – 250 engineers, 5–15 R&D projects annually in one pool of resources, each project has 300–500 tasks. Each day, 15–25 new tasks or other events are identified. Horizon of scheduling is mainly 3–6 months; the schedule is about 5000 interconnected tasks. Criteria: priorities of projects, deadlines, project budget, calendar of engineers availability, competencies of engineers, etc.
- **National Railways - Dispatching Center of the Moscow - Saint Petersburg region:** 49 stations, 48 railways segments (2–4 lines), number of blocks – 3500, 800 trains daily, horizon – 24–36 h, 50 events daily, including maintenance windows, train plan – 40–50 operations. Criteria: minimum average delay, safety intervals, execution of master timetable of trains, time of stops, etc.
- **Moscow Online Store/Delivery of Food:** 18 supermarkets of Auchan, Metro and other chains, 300 orders daily, delivery at the preferred time of the current day (earliest 3 h after order time). Supporting pick-up – workers who consolidate goods for customers; consolidating goods from supermarkets and delivering to customers; changing routes and schedules adaptively "on-the-fly" because of new incoming orders; considering trucks as "mobile storages" in case they are stopped in a traffic jam.

The summary of key complexity factors typical for the considered resource management cases includes: number of daily orders, resources, products and tasks, multi-objective resource management (maximize service quality, minimize costs and time of delivery, maximize profitability), variety of decision makers, which require solving conflicts and harmonization of interests, individual approach to orders and resources, interdependencies between processes and tasks, specifics of order execution such as a dividable orders or alternative processes and resource usage including shared resources, re-usable or consumed resources, recovery of resources, shared costs, etc.

But one of the most important factors of resource management complexity is that in practice humans develop schedules that are not homogeneous - parts of the schedule can differ by applied criteria, preferences and constraints. It reflects the fact that a "good schedule" is usually a well-balanced one among many actors with conflicting interests. However, the achieved balance of interests always depends on the situation – the "optimal" decision becomes event-driven and can change when a new event arrives.

Real time harmonization of users interests requires on-line communication with decision makers, which can not only enter new events, but change their preferences and constraints at any moment, approve or reject decisions and make counter-proposals.

As a result, we consider adaptability as a one of the most important challenges for such applications across all domains which can be defined as the ability to achieve changeable goals under conditions of frequent occurrence of unpredictable events.

3 Overview of Methods and Tools for Resource Management

A number of traditional methods and tools of resource planning and optimization based on linear, dynamic or constraint programming is well known [6, 7].

However, most of these methods and tools are developed for batch process where all orders and resources are given in advance and are not changed in real time.

As a result, in the domain of Enterprise Resource Planning (ERP) classical batch schedulers offered by SAP, Oracle, Manugistic, i2, ILOG, J-Log and other companies still dominate on the market and mainly focused on accounting - resource allocation, planning, optimization and communication have limited use.

To reduce the complexity of combinatorial search and address some of the issues, new methods consider heuristics and meta-heuristics, allowing acceptable decisions in more reasonable time by reducing search options [8, 9]:

- Greedy local search algorithms based on heuristic rules of the domain;
- Artificial intelligence methods, use of neural networks and fuzzy logic;
- Metaheuristics: genetic algorithms, tabu search;
- Simulations including simulated annealing, etc.;
- Stochastic methods such as the Monte Carlo method;
- Ant Colony and Particle Swarm Optimization;
- Combination of parallel heuristic algorithms of optimization, etc.

However, these methods still use batch processing and struggle to take into consideration individual criteria, preferences and constraints, as well as provide real-time adaptation of schedules based on events processing.

Analysis of the above solutions makes it possible to identify the following issues:

- Lack of models, methods and tools for adaptive resource management;
- Solutions require programmers when problem specifications are changed;
- Systems support centralized management based on top-down commands;
- Hierarchical rigidness of systems does not allow for reaction to events;
- Internal passivity and functioning in the batch mode on users' request only;
- Focus on data and not on corporate knowledge;
- Standardization ignores individual preferences of decision makers.

The discussed complexity and high dynamics of business leads to the fact that traditional, centralized, hierarchically organized, sequential methods and algorithms of combinatorial search or heuristics cannot effectively solve the problem of adaptive resource management with the acceptable quality and within the required time.

The search for options for decision making remains very time-consuming, and results are often just not feasible or not comparable to human decisions.

4 New Models and Methods for Adaptive Resource Management: From Optimization – to Consensus

In the last decade, new models and methods of distributed problem solving for resource planning and optimization were developed based on multi-agent technology.

One of most promising approaches is Virtual Market (VM) [10, 11], which got a theoretical basis and has become popular staring from 2010.

The idea of VM is based on ongoing demand and supply matching supported by contract-net protocols: in such multi-agent solutions each agent starts out with some initial set of tasks and then enters into a negotiation process. The negotiation consists of agents repeatedly contracting out assignments among themselves, exchanging tasks as well as money. For the matching assignment, it was proven that global optimum admits an auction-like procedure with tight guarantees.

An idea of using VM models and methods based on self-organization of agents for solving any kind of complex problem looks very attractive for software engineers. Solution of a complex problem is forming here as a competitive equilibrium or consensus of agents, which cannot be improved during computation. Many useful properties of such algorithms are already identified: they are intuitive, can serve individual criteria, preferences and constraints of all participants, provably correct, naturally parallelizable, appropriate for deployment in distributed systems settings and tend to be robust to perturbations of the problem specification.

In our multi-agent developments, we have been using the similar software engineering approach starting from 1999 - having discovered the main positive features of such algorithms in the first multi-agent prototype for Volkswagen factory.

In the next period, the developed multi-agent technology was advanced, following the concept of holonic systems that introduced Agents of Products, Resources, Orders and Staff (PROSA) [12]. In our technology, we made the next step in granularity of agents and introduced Business Process and Task Agents, which form Demand-Resource Networks (DRN), representing self-organized schedules with pro-activity. For agents of DRN we provide new VM method of adaptive decision making based on functions of satisfaction and bonuses-penalties – to provide elasticity by compensations for agents in the processes of solving conflicts and forming consensus [13–15].

During the process of self-organization, DRN agents are at first choosing the best free options, and then resolving conflicts until the system is balanced and none of the options can improve the overall performance of the system.

This process reflects the native way experienced managers and dispatchers usually form complex schedules, solving conflicts and finding a balance of conflicting interests among all parties involved in decision making.

The formalized problem statement and description of method is given in [16].

5 Functionality and Architecture of Solutions

Functionality of developed autonomous AI solutions for adaptive resource management is aimed to support the full cycle of resource management:

- collecting new events via sensors, external systems and mobile devices;
- allocation of orders to resources by matching of the suitable resource;
- scheduling of orders/resources – computing the best possible sequence and detecting time of task beginning and finishing (operations) for orders execution;
- optimization of orders/resources (if time is available) – the ongoing process of improving KPIs of all agents involved in resource management;
- forecasting new events (new orders or failures) which will be processed as virtual events for dynamic pre-reservation of critical resources;
- on-line communication with users: approval of system recommendations, changing preferences or giving counter-proposals, fixing facts, etc.;
- monitoring and control of plan execution – comparing planned and actual results and detecting gaps and triggering re-scheduling event to top-management;
- re-scheduling in case of a growing gap between the plan and reality – for example, if a user is ignoring recommendations and out of given time;
- learning from experience – clustering of events, compare planned and actual time of tasks execution, for example, to analyze productivity of workers;
- real-time "what-if" simulations – in parallel with the main trajectory of plan execution, a few lines of simulations can be running in real time to investigate future;
- evolutional re-design of business network – generating proposals on "how-to" improve quality and efficiency of operations (better place for storage, etc.).

The autonomous AI solution for adaptive resource management is designed as a cyber-physical system [17] integrating physical and cyber components (Fig. 1):

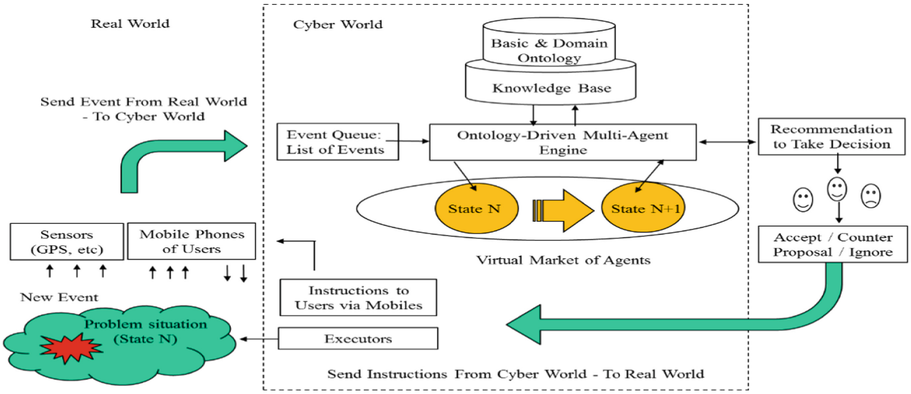


Fig. 1. Cyber-physical architecture of adaptive resource management system

- basic and domain ontology of resource management – to acquire, keep and modify knowledge of problem domain for resource management;
- knowledge base of enterprise – designed as a semantic wikipedia of an enterprise, based on domain ontology which includes information about instances for specific enterprise (so called onto-model of enterprise);
- event queue – works as a buffer and accumulates events which are coming from the real world, regulates policies for event processing and keeps information on the results of events processing;
- ontology-driven multi-agent engine – the generic scheduler customizable by knowledge base of enterprises;
- virtual market (world) of agents – consists of instances of agent classes to be executed, considered as state machines;
- virtual market states (scenes) – represent the previous state (situation) and the proposed changes in schedule at the specific moment;
- sensors, executors, mobile devices and other application components – make it possible to collect events and implement decisions in the real world.

Ontology is aimed for collecting business requirements, customizing the solution for new domain and enterprise specifics and re-programming the solution "on the fly".

Virtual world is considered here as a "co-pilot" model of real world (for example, fleet of trucks is moving on the real roads and in parallel in virtual world on server) which is continuously updated by events and is used for generating decisions and recommendations for users.

The decisions made are forwarded to mobile phones of users or equipment in real world as instructions, but can be re-negotiated when required.

6 Industrial Applications

In the period 2000–2008 we have developed about 15 industrial prototypes and full-scale multi-agent systems for adaptive resource management. Most advanced systems, which can be characterized as an autonomous AI solutions, are presented in Table 1.

Table 1. The first project-based generation of industrial applications

Customer	Project	Results and value for customer
Tankers International	Scheduling of seagoing tankers for one of the largest fleets of Very Large Crude Carrier (VLCC) oil tankers consisting of more than 40 ships and representing just below 10% of the world seagoing tanker capacity	• Reduction of three days of idle runs, per tanker, annually • Taking into account the cost of idle runs per tanker and per day, for 40 tankers the savings generated a return on investment (ROI) of less than six months • Reduced delays of oil deliveries and, consequently, payments of delay penalties • Domain knowledge on running tankers was for the first time collected and formalized in a computer readable format • Dispatchers were provided with scheduling options and delivery price during a telephone conversation with a client • The scheduler typically requires a few seconds to several minutes to complete the analysis
Addison Lee	Taxi scheduling for 2000 GPS-equipped cabs in London and 13 ths. orders daily	• The total number of processed orders increased by 7% within the first month with the same number of resources • 98.5% of all orders were allocated automatically without dispatcher's assistance • The number of lost orders was reduced to 3.5 (by up to 2%) • The number of idle runs was reduced by 22.5% • Each vehicle was able to complete two additional orders per week, which increased the yield of each vehicle by 5–7% • Profitability Increase: +4.8% • Orders collecting time: 40% faster • Time for Dispatcher Training: reduced by 4 times
GIST	Consolidation of cargos for trucks with the use of cross-docks and backhauls	• In the past, 2 operators worked for one day to make a schedule for 200 cargos/now it took 8 min for 200 orders

(*continued*)

Table 1. (*continued*)

Customer	Project	Results and value for customer
		• Before: planning day 1 for day 3/After: planning day 1 for day 2 and even day 1 for day 1 • Before: no support backhauls and consolidations in real time/After: scheduling with backhauls and consolidations • Before: no software to schedule 4000 orders with X-Docks and driver shifts (manual procedure only)/After: 4 h to plan 4000 orders via X-Docks and ability to add new orders dynamically (a few seconds for one order) • Knowledge was hard to share, it was "spread" among experts/Now: capture best practice • Choosing the best route flexibly with the view of consolidation or other criteria
Avis (UK)	Rent-a-car scheduling for one of the UK subsidiaries (250 cars/30 drivers), incl. car washing and fueling, pick-up and delivery, etc.	• Solution processes 15–20 reservations per station with about 80–120 events per hour, including new rentals, events from handheld computers of drivers and from web-users • 90% of all executed decisions have been made 20–30 min before fixing of the schedule for execution • Number of events leading to considerable reorganization of the schedule per day (level of turbulence) ~40% • Reduce of late deliveries – 10%
City Sprint	Couriers/Medical Lab Service	• Reduce of order delays, idle runs and fuel consumption • Possibility to optimize fleet of vehicles
Channel Four	Banners to Web-sites allocation	• Reduce of complexity of banners re-allocations – 15% • Efficiency increase for advertising – 5–10%

Some of designed solutions were used only as a prototypes or decision support tools but some of them got green light for full-scale implementation and work till now.

On the next stage the design of industrial applications was reconsidered and new generation of solutions was developed as products/services in 2010–2017 (Table 2).

Table 2. The second product/service-based generation of industrial applications

Product	Customer	Results and value
Smart Aerospace	Rocket and Space Corporation “Energia”	• System in operation for 8 dispatchers and 120 other users for supplying fuel, water, food, etc • Speeds-up scheduling for flight programs by 4–5 times • Simulation of worst-case scenarios for risk estimates • System is critically important for success of mission
Smart Trucks	Prologics, Lorry, Monopoly, Trasko, Trans-Terminal, etc.	• Increasing the number of orders – up to 3–5% • Reduce of late deliveries – up to 5% • Increase of resource efficiency – 5–10%
Smart Factory	TyazhMash, Axion-Holding, Airbus, AviaAgregat, Irkut, Kusnetsov	Axion-Holding: • Increase of workshop productivity – 5–10% • Savings for one workshop for tools production is 7 man-months monthly
Smart Field Services	Samara Gas Company, Volgograd Water Supply Company, Far East Service Company	Samara Gas Company: • Reduction of reaction time – by up to 5–7 times • Increase of service efficiency – up to 40% (12 orders a day instead of 7 in the past) • Reduction of mistakes of dispatchers
Smart Internet Deliveries of Food	Instamart (Moscow)	• Reduction of assembly time of the order by 15% • Reduction of delays in deliveries to clients by 22% • Client satisfaction is increased: when one courier has a problem, it can be solved with the help of other couriers
Smart Supply Networks	Lego (Chicago outlets), Coca-Cola (Germany), Siberian Coal Mining, “Gaspromneft”	Lego (US): • Increase of profitability – up to 18% Coca-Cola, Germany: • Increase of orders delivered on time – up to 7% • Savings on transport – up to 20%
Smart Projects	Rocket and Space Corporation “Energia”, Ministry of Economics of Samara Region	• Full transparency of projects: from annual plans of delivery in contacts – to daily plans of departments and employees • Reduction of expenses – 5–10% • Increased number of projects implemented within the budget and on time – 15% • Increase in motivation of employees
Smart Railways	Russian Railways (regions of Moscow and Saint-Petersburg, and Siberia-Baykal)	• Reduction of railway train delays – up to 15–25% • Better quality of decisions and easy for dispatchers • Faster reaction on events – 2–3 times • Reduction of human factor • Increase of train speed for Baykal polygon – up to 3–5%

The idea of re-designing of solutions was to form a platform and a number of ready-to-use components and deliver products as a services with minimum investments.

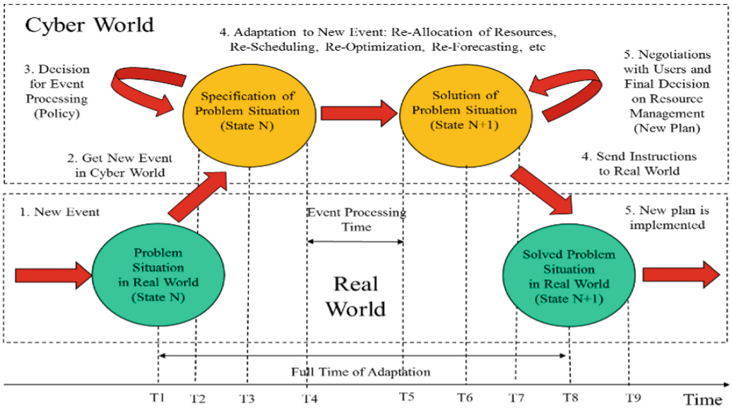


Fig. 2. Life-cycle of event in adaptive resource management

Support of the main decision-making cycle in Autonomous AI System provides opportunity to measure value and time of each event or decision made by user with full transparency for the enterprise on profits and losses (Fig. 2):

- Moment T1: new event appears in real world or at third-party systems;
- Moment T2: new event is specified by the user manually or automatically;
- Moment T3: based on the given policy, the new event will be automatically processed or will wait for specific approval from the authorized decision maker;
- Moment T4: the event triggers resource re-allocation, re-scheduling and re-optimization in the system;
- Moment T5: the new decision is formed and presented to users for approval (or counter-proposal and negotiations: user also can ignore proposal, depending on policy for events), but some decisions can go into real life automatically;
- Moment T6: user feedback is taken into consideration and new adaptive re-scheduling starts with new options for decision making or the decision is approved;
- Moment T7: the approved decision will be sent to the real world: to managers, engineers, technicians, workers or drivers for implementation;
- Moment T8: the decision and new revised plan is executed – products are assembled, transported to customer, etc.;
- Moment T9: the communication session with decision makers is completed.

The value of events or decisions related to transition time (time from one equilibrium to another) gives possibility to measure the Level of Adaptability of the solution.

However, it also helps identify the effect of adaptive resource management vs. the batch ERP approach, f.e., for factories where a plan has monthly granularity (Fig. 3).

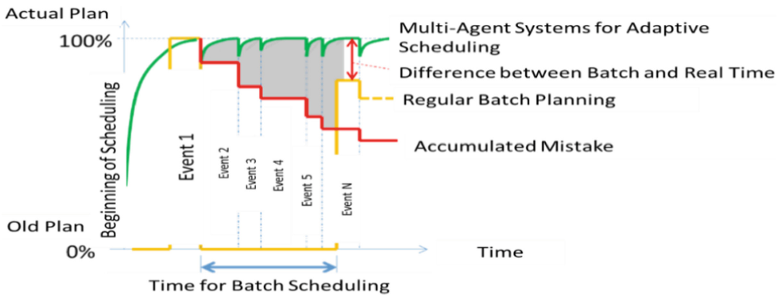


Fig. 3. Value of adaptive resource management for business

Accumulated mistakes of batch planning can generate big losses because of not reacting to a new order, missing deadlines, etc. The estimate of the gap in KPIs between the planned and actual schedules is the key for calculation Return on Investment (ROI).

Another key characteristic of Autonomous Solutions is the number of decisions accepted by users, which can represent the Level of Autonomy.

In our practice, we start with 10–15% of accepted decisions, and then improve it in a step-by-step way up to 50% and more, when the AI system takes more decisions than experienced users.

7 Lessons Learned and Key Benefits

1. Development of the considered systems requires highly qualified domain experts and programmers, is time-consuming, requires a lot of testing, etc.
2. Development of self-organized solutions for business users is challenging: sometimes it is difficult to estimate how far we are from the "optimal" solution; results depend on history of events occurrence (pre-history sensibility); the butterfly effect: small input leads to unexpected big output; system reaction can slow down in case of transition between equilibriums; in case of system re-start, the result of scheduling can be different; it is difficult to "roll back" the system decisions (unreversable); real-time interaction with users becomes sophisticated; system may become too "nervous" during re-scheduling; system decision can be hardly explained to user (loss of causality).
3. Enterprise resource management is critically important for business, and that is why this area is still very conservative in adopting new AI solutions.
4. Big part of enterprise knowledge for decision making is usually hidden and requires direct communication with dispatchers, engineers, workers, drivers, etc. For example, in the "Smart Railways" solution, development started with 2 criteria for decision making (minimum sum of delays and intervals of safety) but after 2 years we had 84 criteria, including such as limitations for cargo trains to transport friable materials on the oncoming track at the time of a high speed train.

5. Significant part of all efforts is associated with web-based user interfaces, which need to be adjustable and low-cost for developments.
6. Dispatchers are often very resistive to innovations and need to be highly motivated to compete with the solution – for example, by bonuses calculated on results achieved.
7. Way forward is to develop SaaS digital platforms for developing an eco-system of services and Add-On solutions which can be integrated with existing systems.

In practice, all these difficulties are manageable but may require special tools for initial analysis of client data or integration with legacy systems.

The discussed difficulties are compensated by the following benefits:

- Allows enterprises to move to real-time economy visualizing profits and losses;
- Improves efficiency of resources by shift to real time decision making;
- Solves complex scheduling problems replacing combinatorial search by analyzing conflicts and finding trade-offs;
- Provides adaptive re-scheduling with fast reaction to events;
- Provides an individual approach to every order, task, product and resource;
- Supports pro-active interactions with users for coordinated teamwork;
- Reduces the role of human factor during decision making;
- Reduces development costs by reusing the code in new applications;
- Enables modeling of "if-then" scenario and forecast to improve decisions;
- Creates a new digital platform to support the growth of business.

The results of developments can be applied for solving wide range of complex problems of resource management in Industry 4.0 and Society 5.0 focused on developing knowledge, talents and social skills of humans.

8 Conclusion

The new class of autonomous AI solutions for enterprise resource management is opening an opportunity to increase the quality and efficiency of business.

The developed industrial applications prove that multi-agent technology provides powerful solutions for solving complex problems of resource management under conditions of high uncertainty, complexity and dynamics. High adaptability of resource management helps to improve efficiency of business, reduce response time and improving quality of service for new orders, avoid loss of orders in peak time, minimize expenses and penalties, improve utilization of resources, etc.

As the next step, we are designing a swarm of multi-agents schedulers for solving extremely complex problems of resource management for manufacturers, railways, etc. Future works will be focused on powering emergent intelligence of autonomous AI solutions by pattern recognition and learning from experience.

This paper has been prepared with the financial support of the Ministry of Education and Science of the Russian Federation – contract №14.578.21.0230, project unique ID is RFMEFI57817X0230.

References

1. Capitalizing on Complexity? Insights from the Global Chief Executive Officer Study, IBM, USA (2010). http://www-01.ibm.com/industries/.government/ieg/pdf/CEOstudy_2010_GovernmentFocus.pdf. Accessed 8 Feb 2018
2. Skobelev, P., Trentesaux, D.: Disruptions are the norm: cyber-physical multi-agent systems for autonomous real-time resource management. In: Borangiu, T., Trentesaux, D., Thomas, A., Leitão, P., Barata Oliveira, J. (eds.) Service Orientation in Holonic and Multi-Agent Manufacturing. SCI, vol. 694, pp. 287–294. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-51100-9_25
3. Gartner: Top Strategic Predictions for 2016 and Beyond: The Future is a Digital Thing (2016). https://www.gartner.com/doc/3142020?refval=&pcp=mpe
4. Artificial Intelligence and Robotics – Imperial College, UK-RAS White Paper, 48 p. (2017). http://hamlyn.doc.ic.ac.uk/uk-ras/sites/default/files/UK_RAS_wp_AI_web_retina.pdf
5. Rzevski, G., Skobelev, P.: Managing Complexity. WIT Press, London-Boston (2014)
6. Leung, J.: Handbook of Scheduling: Algorithms, Models and Performance Analysis. CRC Computer and Information Science Series. Chapman & Hall, London (2004)
7. Voß, S.: Meta-heuristics: the state of the art. In: Nareyek, A. (ed.) LSPS 2000. LNCS (LNAI), vol. 2148, pp. 1–23. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45612-0_1
8. Binitha, S., Siva Sathya, S.: A survey of bio inspired optimization algorithms. Int. J. Soft Comput. Eng. (IJSCE) **2**(2), 137–151 (2012)
9. Rossi, F., van Beek, P., Walsh, T.: Handbook of Constraint Programming. Elsevier Science, New York (2006)
10. Easley, D., Kleinberg, J.: Networks, Crowds, and Markets: Reasoning About a Highly Connected World. Cambridge University Press, Cambridge (2010). http://www.cs.cornell.edu/home/kleinber/networks-book/. Accessed 30 Jan 2018
11. Shoham, Y., Leyton-Brown, K.: Multi-agent Systems: Alghoritmic, Game Theoretic and Logical Foundations. Cambridge University Press, Cambridge (2009)
12. Brussel, H.V., Wyns, J., Valckenaers, P., Bongaerts, L.: Reference architecture for holonic manufacturing systems: PROSA. Comput. Ind. **37**(3), 255–274 (1998)
13. Skobelev, P.: Open multi-agent systems for decision-making support. Avtometriya, J. Sib. Branch Russ. Acad. Sci. **6**, 45–61 (2002)
14. Skobelev, P., Vittikh, V.: Models of self-organization for demand-resource networks, automation and control. J. Rus. Acad. Sci. **1**, 177–185 (2003)
15. Vittikh, V., Skobelev, P.: The compensation method of agents interactions for real time resource allocation. Avtometriya, J. Sib. Branch Russ. Acad. Sci. **2**, 78–87 (2009)
16. Skobelev, P.: Multi-agent systems for real time adaptive resource management. In: Leitão, P., Karnouskos, S. (eds.) Industrial Agents: Emerging Applications of Software Agents in Industry, pp. 207–230. Elsevier, Amsterdam (2015)
17. Leitão, P., Colombo, A., Karnouskos, S.: Industrial automation based on cyber-physical systems technologies: prototype implementations and challenges. Comput. Ind. **81**, 11–25 (2016)

Regular Papers

A Holonic Multi-agent Based Diagnostic Decision Support System for Computer-Aided History and Physical Examination

Zohreh Akbari[1(✉)] and Rainer Unland[1,2]

[1] Institute for Computer Science and Business Information Systems (ICB), University of Duisburg-Essen, Essen, Germany
{zohreh.akbari,rainer.unland}@icb.uni-due.de
[2] Department of Information Systems, Poznan University of Economics, Poznan, Poland

Abstract. Available Medical Diagnosis systems (MDSs), including the state of the art, mainly focus on finding the perfect link between the patient's medical history and their health knowledge. However, no matter how powerful they are in performing this action, it is always possible that the final strong deduction is based on some incomplete input. Prior to this process, a physician should literally perform the Differential Diagnosis (DDx), in which (S)he carefully listens to the symptoms explained by the patient, considers some potential diagnoses and then tries to gather enough evidence and supporting information to shrink the probability of the other candidates. In a patient encounter, this method is used in a process called the History and Physical examination (H&P). Only physicians and in some institutions, in order to compensate the shortage of the physicians, some specially trained nurses are qualified to perform this process. A system capable of guiding a focus H&P, however, will allow less experienced nurses to perform this process, and furthermore, can provide second opinions in critical cases. The DDx domain is in fact a holonic domain; hence, a MDS with holonic architecture could be able to perform this process. As the Holonic Medical Diagnosis System (HMDS), tends to cover the stages in the H&P, this system could also be added to available MDSs in order to provide them with the essential comprehensive input and allow their integration in the clinical workflow. This paper will demonstrate the performance of this system and concentrates on its learning process.

Keywords: Holonic Multi-Agent System (HMAS) · Machine Learning (ML) Medical Diagnosis System (MDS) · Reward engineering

1 Introduction

A Diagnostic Decision Support Systems (DDSS) or Medical Diagnosis System (MDS) is a specific type of Clinical Decision Support Systems (CDSSs) that is developed to provide an ordered list of potential diagnoses for given signs and symptoms. The physician then takes the suggested diagnoses and the supportive information into

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 29–41, 2018.
https://doi.org/10.1007/978-3-319-94580-4_3

consideration, and if necessary orders further tests to narrow down this list [1]. In fact, these systems can save the physician's time, and remind ignored critical possibilities.

Available systems, including the state of the art, mainly focus on finding the perfect link between the given input and their health knowledge. However, prior to this process there should be a precise method to guide the user in providing the right, all-encompassing input. This is similar to what a physician does when listening to a patient. (S)he would carefully listen to the symptoms explained by the patient, considers some potential diagnoses and then tries to gather enough evidence and supporting information to shrink the probability of the other candidates. This method is called the Differential Diagnosis (DDx). In medicine, a DDx is the distinguishing of a particular disease or condition from others presenting similar symptoms [2], which helps to reduce diagnosis errors [3]. In a patient encounter, this method is used in a process called the History and Physical examination (H&P). The H&P is a critical component of a patient encounter in which information relevant to a present complaint is obtained [4]. The H&P report includes the Chief Complaint (CC), the Medical History (Hx), the Physical Exam (PE) and the Assessment and Plan (DDx & P). As mentioned, the state of the art of the MDSs merely covers the last part and takes the result of the other parts as input. This is exactly one of the limitations of these systems, and the reason why they cannot be well integrated into the clinical workflow of the hospitals.

The state of the art of the MDSs includes the IBM Watson Health [5] and the Isabel [6]. The IBM Watson Health is claimed to be the most powerful Artificial Intelligence (AI) based system capable of performing medical diagnosis. This system is literally a cognitive computer system that has the capability of understanding the natural language. Although Watson is very powerful, its success is very much depending on the quality of the input, i.e. the output of already performed H&P, and it is actually best used in order to create patient-specific treatment plans. Literally Watson has never been involved in the medical diagnostic process, but only in improving the diagnosis and assisting with identifying treatment options for patients who have already been diagnosed [7]. Isabel on the other hand, is a knowledge-based CDSS that facilitates diagnostic reminders and DDx [6]. A systematic review of DDx generators was conducted in [8], according to which Isabel was associated with the highest rates of diagnosis retrieval. However, as stated in [9] Isabel is still too slow and its accuracy drops significantly if only limited information is available [10].

As mentioned, no matter how powerful the MDSs are in connecting the patient's medical history to medical knowledge; it is always possible that the final strong deduction is based on some incomplete input. It is also discussed in [11] that: "No matter how good you are at diagnosing and treating, unless you asked the right questions in a timely manner, all the knowledge in the world won't be helpful." Hence, a focused H&P is the key to a flawless diagnosis. As the shortage of physicians is worsening in the recent years, roles such as Physician Assistant (PA) and Nurse Practitioner (NP) have been introduced in order to ease the problem. PAs and NPs are qualified to perform the H&P step, diagnose medical problems and carry out necessary treatments mainly under the supervision of a physician. Undertaking the H&P step, they would help the doctors to be able to see more patients within a certain period of time, as they would just need to review and asses the already prepared H&P report. A system capable of guiding a focus H&P, however, will allow less experienced nurses to perform this process and can also provide second opinions in critical cases. Such system could also be added to available MDSs in order to provide them with the essential comprehensive input.

As the DDx domain meets the characteristics of holonic domains (see Sect. 2), the general research question here is whether a holonic multi-agent architecture could in practice support the implementation of the DDx. Hence, in fact, the development of a Holonic Multi-Agent System (HMAS), which is capable of performing DDx, is the practical contribution of this work. On the other hand, as is the case with intelligent systems, the introduction of the Machine Learning (ML) techniques that support the functionality of this system, is the theoretical contribution of this work. In this regard, the sub-question being answered in this paper is how we can provide the system with the right learning data and more particularly unbiased feedback.

Hence, the rest of the paper is organized as follows. Section 2 briefly describes the holonic nature of DDx, and in Sect. 3, the Holonic Medical Diagnosis System (HMDS) is introduced (for a more comprehensive description of the system please refer to [12]). Section 4 describes the ML techniques that have been adapted and applied to the system, and Sect. 5 concentrates on the rewarding process. Section 6 concludes with a summary and some directions for the feature researches.

2 The Holonic Nature of DDx

The domain attributes that indicate the appropriateness of a multi-agent based solution are presented in [13]. These criteria are clearly met by the DDx problem. Firstly, the environment is open, dynamic, uncertain and complex. Moreover, considering the doctors with different specialties and subspecialties, a MDS can be naturally modelled as societies of cooperating agents, and hence here the agents are natural metaphors. This also implies the distribution of data, control and expertise.

An interesting overview of Multi-Agent System (MAS) architectures is presented in [14]. One of the well-known architectures is the HMAS architecture. The term holon was first introduced in [15] in order to name recursive and self-similar structures. A holonic agent of a well-defined software architecture may join several other holonic agents to form a super-holon, which is then seen as a single holonic agent with the same software architecture [16]. A holon can be a member of several super-holons, if the super-holons' goals do not contradict each other or the holon is indifferent to those conflicting goals [16]. The organizational structure of a holonic society, i.e. the holarchy, offers advantages such as robustness, efficiency, and adaptability [16]. Obviously, all these advantages do not mean that this architecture is more effective than the others, however, specifically for the domains, it is meant for. As stated in [16] such domains should (1) involve actions of different granularities, (2) induce abstraction levels, (3) demand decomposable problem settings, (4) require efficient intra-holonic communications, (5) include cooperative elements, and (6) may urge real-time actions.

The DDx problem can be decomposed recursively into subproblems of weighting the probability of the presence of the possible diseases. These subproblems may induce different abstraction levels and can be of different granularities. According to the nature of DDx, the problem solvers are collaborative and those dealing with similar diseases need to have more communications, which are to be conducted in timely manner. Hence, this domain meets the characteristics of holonic domains. Another attractive characteristic of HMASs is the support for self-organization, which is the autonomous continuous arrangement of the parts of a system in such a way that best matches its objectives. The HMDS also takes advantage of this characteristic (see Sect. 4).

3 The Holonic Medical Diagnosis System (HMDS)

The HMDS, first introduced in [17], is a HMAS, consisting of medical experts (holons). This structure is applied to this system in a way that supports DDx.

3.1 The Architecture of the HMDS

In general, a MDS either rely on highly smart deliberative agents as one extreme or on a large set of comparatively simple (reactive) agents as the other extreme. The HMDS as a HMAS realizes an improved version of the latter approach. It consists of two types of agents: comparatively simple Disease Representative Agents (DRA) as the system fundamentals on the lowest level and more sophisticated Disease Specialist Agents (DSA) as decision makers on the higher levels of the system [12] (see Fig. 1).

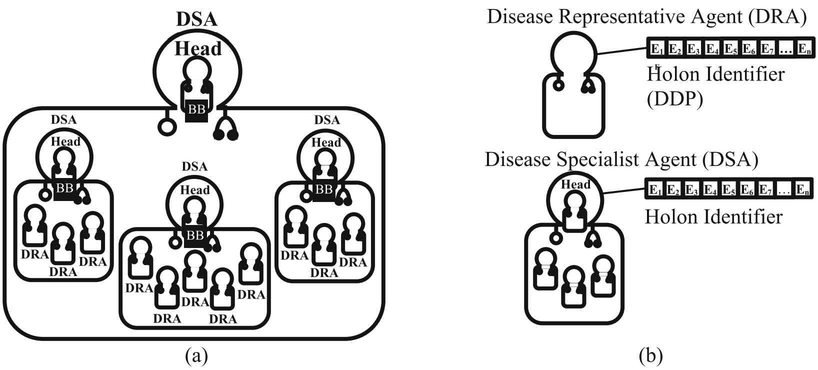


Fig. 1. (a) DRAs and DSAs in HMDS (BB: Blackboard) (b) Holon identifier

DRAs are atomic agents, and form the leaves of the holarchy. Each DRA is an expert on a specific disease and maintains a pattern store that contains the Disease Description Pattern (DDP) – an array of possible signs, symptoms, and test results, i.e. the holon identifier. Thus, in order to join the diagnosis process, these agents only need to perform some kind of pattern matching, i.e. calculating their Euclidean distance to the diagnosis request. DSAs are holons consisting of numbers of DRAs and/or DSAs with similar symptoms; i.e., representing similar diseases. This encapsulation, in fact, enables the implementation of the DDx. The DSAs are designed as moderated groups [16], where agents give up part of their autonomy to their super-holon. To this end, for each DSA, a head is defined, which provides an interface and represents the super-holon to its environment. This head is created for the lifetime of the super-holon, based on agent cloning [18]. For this purpose, each head is capable of cloning, i.e., creating a copy of its code, and passing the relevant information to the new agent [12]. In this instance, the holon identifier will be the average of the holon identifiers of the members.

The holarchy has one root, in fact a DSA, which will play the role of the common and exclusive interface to the outside world. The system starts with this DSA, takes all the DRAs as its members, and then let the DSAs form automatically. Although this process is based on the affinity and satisfaction [19], and at the beginning, no information about the satisfaction factor is available, it is still possible to initially form the DSAs based on the affinity, i.e., the similarity between the diseases[1]. To this end, the mentioned DSA accepts the DRAs, as its members, clusters them, and defines for each of the clusters, i.e. super-holons, a head. This is repeated recursively until no further clustering is necessary (see Sect. 4.1). This can be performed once as the system is being defined and accelerate the self-organization. Later on, the system can still refine its architecture using its self-organization technique (see Sect. 4).

3.2 The Functionality of the HMDS

In principle, the HMDS works as follows: The head of the holarchy receives the diagnosis request as a specific combination of signs, symptoms and medical test results and places it on its blackboard[2]. Each agent of the system that has knowledge of this blackboard, i.e. any member of this super-holon, can read the messages on this blackboard. A DRA's reaction to a request message is to send back its similarity to the request. However, based on the provided information a DSA can decide whether it wants to join the diagnosis process or not. This will actually control the data flow in the holarchy. This decision is made based on some simple statistical information about the DSA's members. Considering the holon identifier of the members, the head will check if the request would be an outlier, and in negative cases it will decide to join the diagnosis process. This means that it will read the information from its head's blackboard and will place it on its own blackboard. Then the same process repeats recursively until the request reaches the final level of the holarchy. Results obtained by participating agents now flow the other way round from bottom to the top of the holarchy and are sorted according to their similarity and frequency. Literally, each agent will send its top diagnoses together with all of the signs, symptoms or test results that are relevant from its point of view, but their presence or absence has not been specified. This implies that originally not

[1] The Euclidean distance between the DDPs of the DRAs.

[2] In HMDS, the communications are solely done via the blackboard systems [30] of the DSAs.

provided relevant information might be requested from the user in a second step [12]. Section 3.3 demonstrates the functionality of the system.

3.3 The Examples of the System Simulations

The simulations presented in this paper have been conducted using the GAMA platform, and the disease-related data have been gathered from Mayo Clinic website [20]. For the following cases, a small system with only 20 diseases has been simulated.

Case 1 (Diagnosis of Lung Cancer). The first example is based on an actual H&P report, which is provided in [21] for study purposes. In our simulation, a holarchy with three levels is formed. The chief complaint is entered into the system and the system's reaction is monitored using the actual report as a benchmark. The H&P report includes:

Chief Complaint (CC): Shortness of Breath (SOB)
History of Present Illness (HPI): Chest pain, Chills, Cough, Fever, History of breathing troubles, History of cancer in family, History with tobacco, Night Sweats, Productive cough, Vomit, Weight loss, Wheezing
Review of Systems (ROS): Anxiety, Fainting, Fatigue, Weakness, Heart palpitations and arrhythmias, Changes in skin color, Changes in appetite
Physical Examination (PE): Cyanosis, Edema, Swollen lymph nodes
Diagnostic Tests: Blood test, CT scan
Assessment and Plan: (1) Asthma (Asthma tests), (2) Lung Cancer (X-ray/CT scan, Biopsy), (3) Pneumonia (Blood Test), (4) Sarcoidosis (Blood Test), (5) Tuberculosis (PPD: Purified Protein Derivative skin test for tuberculosis).

In HMDS, entering the SOB, the DSA of the pulmonary diseases will be activated and the suggested signs and symptoms to be checked includes: Anxiety, Chest discomfort, Chills, Cough, Cyanosis, Diarrhea, Fainting, Fatigue and Weakness, Fever, Presence of frequent respiratory infections, Heart palpitations and arrhythmias, Hoarseness, Itching, Loss of appetite, Nausea and vomit, Night sweats, Phlegm, Sweats, Edema, Swollen lymph nodes, Weight loss, and Wheezing. These signs and symptoms very much match the ones mentioned in HPI, ROS and PE sections of the original H&P report. After entering the value of these signs and symptoms according to their presence or absence, the final DDx list would be: (1) Asthma, (2) lung cancer, (3) Pulmonary Edema, (4) Tuberculosis, (5) Sarcoidosis, (6) Pneumonia, (7) Bronchitis, (8) Pulmonary Embolism, (9) COPD, and (10) Lymphoma. The suggested medical tests would be asthma tests, x-ray/CT scan, sputum cytology, biopsy, pulse oximetry, arterial blood gas analysis, and sputum test for tuberculosis. These results match the actual H&P to a considerable degree and could even be improved through learning. In this case, the CT scan showed a large mediastinal mass, and the final diagnosis was lung cancer.

Case 2 (Metastatic Lung Cancer to Common Bile Duct Cancer). The HMDS also acts well in the presence of multiple diseases, e.g. the metastasis cases. This example is extracted from a medical paper in [22]. The signs and symptoms in this case included abdominal pain, coarse breath sounds, dry cough, jaundice, and shortness of breath; and the final Diagnosis was metastatic lung cancer to common bile duct cancer, with suggested medical tests to be: Blood test, CT scan, ERCP (Endoscopic Retrograde Cholangiopancreatography), and Biopsy. Giving these symptoms to the HMDS two different DSAs will be activated: the DSA of pulmonary diseases and the DSA of Hepatology and Gastrointestinal Disorders. Their super-holon will then put the output of both members in order. The DDx list will be: (1) Bile Duct Cancer, (2) Cholangitis, (3) Asthma, (4) Lung Cancer, (5) Hepatitis B, (6) Pulmonary Edema, (7) PSC, (8) Pulmonary Embolism, (9) Bronchitis, (10) Lymphoma. This DDx list actually includes the bile duct cancer as the first and the lung cancer as the forth possible diagnosis, and therefore the possibility of Metastasis can be clearly mentioned to the doctor.

4 Learning in HMDS

As mentioned in Sect. 3.1, the initial holarchy of the system can be created using clustering in different levels of the holarchy. Clustering is an unsupervised learning technique and hence doesn't require any learning feedback. After having the initial holarchy, however, it is still essential to support the system in learning and updating based on the new observations. Medical knowledge demonstrates a steady upward growth, and diagnosis is also very much affected by the geographical regions. Hence, in order to adapt and improve the behavior of the system, we need to: (1) Update the medical knowledge based on the new instances, (2) Refine the holarchy according to the experience and the feedback. In HMDS, holon identifiers are updated applying the exponential smoothing method, as a supervised learning method, and the self-organization of the holarchy is supported by Q-learning, as a reinforcement learning technique. Since both methods require feedback from the environment, it is clear that the quality of the feedback is of central importance. The rest of this section covers the above mentioned techniques and provides an experimental example, however, the discussion on the learning feedback can be followed in Sect. 5.

4.1 The Machine Learning Techniques Used in HMDS

Clustering. The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [23] is one of the best algorithms for clustering in HMDS. In [24] a simple and effective method for automatic detection of the input parameter of DBSCAN is presented, which helps best to deal with the complexity of the problem at hand.

Exponential Smoothing. Exponential smoothing is a very popular scheme for producing smoothed time series [25]. Using this technique, the past observations are assigned exponentially decreasing weights and recent ones are given relatively higher weights:

$$s_t = \alpha . x_t + (1 - \alpha) . s_{t-1} \quad (1)$$

where α is the smoothing factor, and $0 < \alpha < 1$. As it is clear, the smoothed statistic s_t is a simple weighted average of the current observation x_t and the previous smoothed statistic s_{t-1}. In HMDS, this method is used in order to update the holon identifiers.

Holonic-Q-Learning (HQL). The Holonic-QL is a Q-learning technique introduced for self-organization in the HMDS [12]. In HQL, the Q-value is in fact measuring how good it is for a holon to be a member, of another holon. In this case, the states are the existing holons $\{h_i\}$ and action h_i indicates becoming a sub-holon of holon i:

$$\begin{aligned} Q_t(sub(h),h) \leftarrow\ & (1-\alpha_t)(1-\alpha_t)Q_{t-1}(sub(h),h) \\ & + \alpha_t(R_t(sub(h),h) + \gamma \operatorname{argmax}_{Q_{t-1}(h,\mathrm{sup}(h))}(Q_{t-1}(h,sup(h)).Aff(sub(h),sup(h)))) \end{aligned} \tag{2}$$

where, $\alpha_t = \frac{1}{1+visits_t(sub(h),h)}$, $\gamma \in [0,1)$ is the discount factor, $Aff(sub(h),h) = 1 - \frac{d(sub(h),h)}{\max d(sub(h),h)}$, and the reward is calculated by the head of a super-holon (Sect. 5). For more information regarding the HQL and the proof of convergence please refer to [12].

4.2 An Example of Learning in HMDS

Case 3 (A New DDx for Arthritis). In this simulation, the system covers 45 diseases in a holarchy with four levels. Here, again, a real case is used, in which the Madelung-Launois-Bensaude disease (MLB) is suggested as a new DDx for arthritis [26]. MLB is a disease that causes the concentration of fatty tissue in proximal upper body. In 2008, for the first time some instances of this disease have been observed with distal fatty tissue that were misdiagnosed first as arthritis, which normally includes joint pain and joint swelling. In this experiment, the new observations will be given to the HMDS, which so far does not consider the MLB disease with arthritis for the reason of DDx. The system should then be able to come to the same conclusion as in [26] and add the DRA acting for the MLB disease to the super-holon containing the DRA acting for arthritis. Essentially, if an agent is not involved in a diagnosis process it will not receive any reward and hence its Q-value will remain same during that round. As the agent participates in a diagnosis process, it will be rewarded and hence its Q-value will be updated. In case the Q-value of any of the members of the super-holon is getting close to be a noise (close to lower three-sigma limit), the agent will start exploring new opportunities to join new super-holons. One promising approach for this agent is to try to become a member of those super-holons that were activated at the same time with its current super-holon. This will guarantee that the agent would have some common interests with the members of its new super-holon(s).

Figure 2 shows the changes in the Q-values of the different DRAs in case 3. Entering the distal MLB instances into the system, the Q-value of the DRA acting for the MLB disease will get closer to the lower outlier threshold. At this point, the super-holon of the arthritis disease will also be activated and as the DRA acting for the MLB disease is looking for a chance to join some new super-holons, it will be informed and will try to become a member of this super-holon. Considering the holon identifier of its members, this super-holon will then check whether the DRA acting for the MLB disease would be an outlier and since the case is negative, it will accept this new

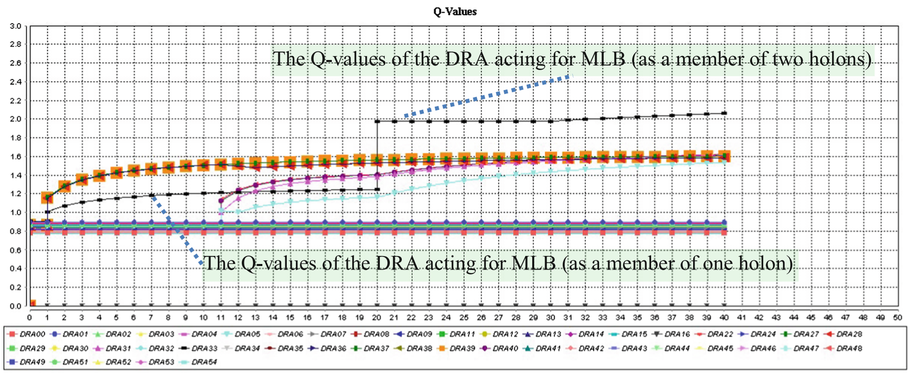


Fig. 2. Changes in the Q-values of the different DRAs in case 3

member. At this stage, the Q-value will be an accumulated value, calculated based on the rewards received from both super-holons. As it can be followed on the diagram, since this value is now greater than the average of the Q-values in both super-holons, the DRA acting for MLB will stop exploring at this point.

5 Reward Engineering in HMDS

It should be noted that the terms "feedback" and "reward" have different definitions here. A feedback is the final diagnosis, suggested by the physician for a given diagnosis request. However, a reward is a numerical value, which is calculated using a reward function that considers the feedback. Agents receive their rewards from their environments. The HMDS has distributed environments, i.e. the highest holon receives the feedback from the outside world, however, the environment for the rest of the holons is their own super-holon, from which they receive their rewards.

In the HMDS, the system will announce the result in form of a list of top 10 possible diseases, and will then receive a feedback, i.e. the final diagnosis, from the physician, and would consider the physician diagnosis for the reason of reward calculation. However, the system cannot simply take this diagnosis to calculate the rewards, but should have a strong policy to reduce the possibility of errors and biased diagnoses. The system is meant to be used in hospitals, and hence, is actually multi-user and receives the feedback from different physicians. Moreover, the system will consider counterfactual learning, which refers to the ability to learn from forgone outcomes, i.e. the outcome of the option(s) that were not chosen [27]. The use of this method literally improves decision-making when evidence is solid but causal links are unclear [28]. In order to design a mechanism to consider the other options next to the physician's final diagnosis, the probability of selecting the options is estimated with a combination of ε-greedy and softmax rules. Using this method the physician's diagnosis is given the highest selection probability, i.e. $1 - \varepsilon$, and all the others are ranked and weighted using the softmax rule. For this reason, we have used the most common softmax method, which uses a Gibbs or Boltzmann distribution, and have adapted it to our problem as follows:

$$p(d_i) = \varepsilon \frac{e^{simF_i/\tau}}{\sum_{j=0}^{9} e^{simF_j/\tau}} \tag{3}$$

where d_i for $0 \leq i \leq 9$ is a disease in the output list, $simF_i$ is the product of the similarity of d_i to the diagnosis request[3] and its frequency, and τ is a positive number called temperature. Low temperatures cause a greater difference in the selection probability of the diseases.

After clarifying the problem with the feedback, next step will be to design the reward protocol. The term "reward engineering", first coined by Dewey in [29], refers to the engineering of the agent's environment in order to make the reward assignment more reliable. According to [29], in reward engineering the agent's goal is not changed, however, the environment is being partially designed so that reward maximization leads to desirable behavior. For this reason, the human factor may even be removed from the loop and the reward may be assigned via an automatic mechanism.

In the HMDS, the holon which is on the top level of the holarchy will receive the final diagnosis[4] and will place it on its blackboard. Consequently, all the members will be informed and the DSAs would repeat the same action, i.e. place the final diagnosis on their blackboard, until the announcement reaches the DRAs[5]. The DSAs, which have participated in the diagnosis process, are now responsible for calculating the rewards for each of their members. In such a super-holon, the DRAs and the DSAs, which have not participated in the diagnosis process, will respond to their head by sending back their similarities to the final diagnosis. The DSAs, which have originally participated in the diagnosis process, however, will respond by sending back the highest reward they have given to their members. These values are called the suggested rewards. Considering the suggested rewards, the super-holon will take the highest value as its highest reward and then assign rewards to its members based on the three-sigma rule:

$$r_i = \begin{cases} 0, & suggR_i \text{ } is\ an\ outlier \\ r^*(3\sigma - |\bar{r} - suggR_i|)/3\sigma, & else \end{cases} \tag{4}$$

where $suggR_i$ is the actual value suggested by the $i-th$ member, $\bar{r}$ is the mean of the suggested rewards, r^* is the highest reward, and σ is the standard deviation of the suggested rewards. Hence, the value of the rewards, except for the case of a penalty, is a fraction of the r^* (always between zero and one), and implicitly considers the

[3] The Euclidean distance between the corresponding DDP and the diagnosis request.

[4] The holon identifier of the DRA acting for the final diagnosis, which is already updated considering the new observation.

[5] If a super-holon, which has originally not participated in the diagnosis process, realizes that the disease given as the final diagnosis matches one of its members, it will assign the reward −1 (penalty) to this member.

satisfaction and affinity factors both. Greater reward values indicate that the problem is more relevant to the super-holon, and thus incompatibilities will be penalized more. Hence, in case a member of a super-holon is close to be a noise, its assigned reward and consequently Q-value will differ greatly from the average rewards and Q-values. This would then lead to higher probabilities of exploration (An agent will decide to explore new opportunities if its Q-value is close to be an outlier in its super-holon).

6 Conclusion

This paper explained that one of the main limitations of the available MDSs is the lack of support for the H&P examination, which prevents their integration in the clinical workflow. The H&P examination uses the DDx method and this domain is in fact a holonic domain, hence, a MDS with holonic architecture could be able to perform the DDx process. The HMDS tends to cover the stages in the H&P process, and hence allow less experienced nurses to perform this process, and can also provide a second opinion in critical cases. This research also discussed that this system should be supported with appropriate ML techniques in order to maintain and improve its functionality. Along with the learning methods, the quality of the learning data, i.e. the feedback is also of central importance. In HMDS, the final report is given to a physician and s(he) makes the final diagnosis, however, in rewarding stage the system cannot simply take this diagnosis, but should have a strong policy to reduce the possibility of errors and biased diagnoses. For this purpose, the system, which is meant to be used in hospitals is actually considered as a multi-user system and hence receives the feedback from different physicians. Moreover, it will apply counterfactual learning and considers the option(s) that were not chosen. This paper suggests a combination of ε-greedy and softmax rules for this purpose. As demonstrated in this paper, the simulation of the system is now in progress and future work will include the complete simulation and validation of the system based on the reward engineering proposed in this work.

References

1. Berner, E.: Clinical Decision Support Systems: Theory and Practice. Springer, New York (2016). https://doi.org/10.1007/978-0-387-38319-4
2. Merriam-Webster: Differential Diagnosis. https://www.merriam-webster.com/dictionary/differential%20diagnosis
3. Maude, J.: Differential diagnosis: the key to reducing diagnosis error, measuring diagnosis and a mechanism to reduce healthcare costs. Diagnosis **1**(1), 107–109 (2014)
4. Segen, J.: Concise Dictionary of Modern Medicine. McGraw-Hill, New York (2006)
5. IBM Watson Health. https://www.ibm.com/watson/health/
6. Fisher, H., Tomlinson, A., Ramnarayan, P., Britto, J.: ISABEL: support with clinical decision making. Pediatr. Nurs. **15**(7), 34–35 (2003)
7. Saxena, M.: IBM Watson Progress and 2013 Roadmap. https://www.slideshare.net/manojsaxena2/ibm-watson-progress-and-roadmap-saxena/7-Watson_Healthcare_Products_1H_2013. Accessed 22 Mar 2018

8. Riches, N., Panagioti, M., Alam, R., Cheraghi-Sohi, S., Campbell, S., Esmail, A., Bower, P.: The effectiveness of electronic differential diagnoses (DDX) generators: a systematic review and meta-analysis. PLoS One **11**(3), e0148991 (2016)
9. Yuan, M.: Watson and healthcare: how natural language processing and semantic search could revolutionize clinical decision support. https://www.ibm.com/developerworks/library/os-ind-watson/. Accessed 22 Mar 2018
10. Graber, M.L., Mathew, A.: Performance of a web-based clinical diagnosis support system for internists. J. Gen. Intern. Med. **23**(1), 37–40 (2008)
11. Husain, I.: Why IBM's artificial intelligence "Watson" could not replace a physician. In: iMedicalApps. https://www.imedicalapps.com/2011/02/ibm-watson-replace-physician-artificial-intelligence/. Accessed 22 Mar 2018
12. Akbari, Z., Unland, R.: A holonic multi-agent system approach to differential diagnosis. In: Berndt, J.O., Petta, P., Unland, R. (eds.) MATES 2017. LNCS (LNAI), vol. 10413, pp. 272–290. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64798-2_17
13. Wooldridge, M.: Chapter 9: Methodologies. In: An Introduction to MultiAgent Systems. Wiley (2009)
14. Lavendelis, E., Grundspenkis, J.: Open holonic multi-agent architecture for intelligent tutoring system development. In: Proceedings of the IADIS International Conference on Intelligent Systems and Agents (2008)
15. Koestler, A.: The Gost in the Machine. Hutchinson, Paris (1967)
16. Gerber, C., Siekmann, J., Vierke, G.: Holonic multi-agent systems. Technical report DFKI-RR-99-03, German Research Centre for Artificial Intelligence (1999)
17. Unland, R.: A holonic multi-agent system for robust, flexible, and reliable medical diagnosis. In: Meersman, R., Tari, Z. (eds.) OTM 2003. LNCS, vol. 2889, pp. 1017–1030. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39962-9_97
18. Shehory, O., Sycara, K., Chalasani, P., Jha, S.: Agent cloning: an approach to agent mobility and resource allocation. IEEE Commun. Mag. **36**(7), 58–67 (1998)
19. Rodriguez, S.: From analysis to design of holonic multi-agent systems: a framework, methodological guidelines and applications. Ph.D. thesis, University of Technology of Belfort-Montbéliard (2005)
20. Mayo Clinic. https://www.mayoclinic.org/
21. University of North Carolina - School of Medicine, History and Physical Examination (H&P) Examples. https://www.med.unc.edu/medclerk/education/grading/history-and-physical-examination-h-p-examples
22. Cochrane, J.: Metastatic lung cancer to the common bile duct presenting as obstructive jaundice. J. Hepatol. Gastrointest. Disord. **2**(121) (2016). https://www.omicsonline.org/open-access/metastatic-lung-cancer-to-the-common-bile-duct-presenting-as-obstructivejaundice-jhgd-1000121.php?aid=69570
23. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Simoudis, E., Han, J., Fayyad, U. (eds.): Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-1996), pp. 226–231 (1996)
24. Akbari, Z., Unland, R.: Automated determination of the input parameter of the DBSCAN based on outlier detection. In: Artificial Intelligence Applications and Innovations, IFIP Advances in Information and Communication Technology, vol. 475, pp. 280–291 (2016)
25. NIST/SEMATECH e-Handbook of Statistical Methods. http://www.itl.nist.gov/div898/handbook/. Accessed Jan 2018
26. Lemaire, O., Paul, C., Zabraniecki, L.: Distal Madelung-Launois-Bensaude disease: an unusual differential diagnosis of acromalic arthritis. Clin. Exp. Rheumatol. **26**, 351–353 (2008)

27. Palminteri, S., Lefebvre, G., Kilford, E., Blakemore, S.-J.: Confirmation bias in human reinforcement learning: evidence from counterfactual feedback processing. PLoS Comput. Biol. **13**(8), e1005684 (2017)
28. Kayes, D., Anna, K.: The Learning Advantage: Six Practices of Learning-Directed Leadership. Springer, London (2011). https://doi.org/10.1057/9780230305595
29. Dewey, D.: Reinforcement learning and the reward engineering principle. In: 2014 AAAI Spring Symposium Series (2014)
30. Corkill, D.: Blackboard systems. AI Experts **6**(9), 40–47 (1991)

A Resilient Agent-Based Re-organizing Traffic Network for Urban Evacuations

Mohammad Al-Zinati[1(✉)] and Rym Zalila-Wenkstern[2]

[1] Jordan University of Science and Technology, Irbid 22110, Jordan
mhzinati@just.edu.jo
[2] University of Texas at Dallas, Richardson, TX 75080, USA
rymw@utdallas.edu

Abstract. Implementing effective traffic road reversals is a complex problem: it requires clearing roads from traffic before implementing safe road reversal operations and often results in anomalies in the network topology. Road reversals are further complicated when, due to unexpected events (e.g., torrential rains), roads are suddenly closed. Current traffic road reversal approaches are based on the execution of mathematical models which identify upfront, optimal reversal configurations for the entire traffic network. These approaches assume that the traffic network structure is static, and as such do not allow for dynamic road closures. In this paper, we present a *resilient* agent-based re-organizing traffic model for urban evacuations. Resilience refers to the traffic network's ability to regain its evacuation function quickly and efficiently after severe perturbations. The proposed model integrates *road reversal* and *zoning* strategies. Experimental results show that: (a) the model improves the evacuation effort, and (b) the evacuation function is able to cope quickly and effectively with dynamic road closures.

Keywords: Multiagent simulation · Resilience · Evacuation

1 Introduction

Natural disasters such as hurricanes, tsunamis and earthquakes are on the rise, and hundreds of millions of people are affected world-wide each year. Effective planning to flee a city is critical to reducing disaster-related casualties. Dealing with disaster, small or large, involves *managing movement*: moving people away from danger, transporting casualties to medical facilities, delivering supply to support rescue operations, etc. Managing "evacuation movement" is often done through a transportation network that has been damaged or is loaded beyond capacity. Conventional approaches for urban evacuation have been proposed [13, 24]. These approaches are based on mathematical models which identify upfront the evacuation strategy for the entire traffic network, often considered to be static. As smart cities are becoming a reality, technologies known as Intelligent Transportation Systems (ITS) have been considered as viable solutions for the definition of *adaptive* evacuation operations.

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 42–58, 2018.
https://doi.org/10.1007/978-3-319-94580-4_4

Several ITS strategies for urban evacuation have been investigated. Among these, researchers have proposed the use of road reversal operations as a mean to efficiently utilize the traffic network capacity [11,21,25]. Despite their theoretical effectiveness, and due to their additional implementation cost, control, and safety issues, road reversal operations are still not widely used in practice [20,25]. The implementation of road reversals is further complicated when the traffic density reaches critical levels [23], and the road infrastructure is damaged unexpectedly. These factors result in a degradation of the evacuation effort.

The ability of a traffic network to quickly restore its evacuation operation using the part of the network not affected by damages, is known in the traffic literature as *resilience* [17,19]. Although there is an increasing interest in investigating resilience [8,15], none of the existing works studies the use of road reversal strategies to dynamically restore the evacuation operation.

In this paper, we present a model for resilient agent-based traffic networks for urban evacuation. The proposed model has the following characteristics: (1) it maintains a resilient network topology that is re-organized dynamically to recover evacuation operation following unexpected road closures; (2) it is based on a combination of *road reversal* and *zoning* strategies. (3) it considers infrastructure-to-infrastructure communications.

The model is defined in the context of the Agent-based Transportation System (ATS) model [27]. ATS is a multi-layered, integrated ITS which consists of *micro*- and *macro*-level agent-based elements; micro-level elements include *vehicles* and *intersection controller agents* and macro-level elements consist of specialized agents called *zone* and *traffic managers*. A *zone manager* is responsible for informing, monitoring, and guiding micro level agents within its zone to ensure that micro-level behaviors and interactions are consistent with the global system behavior. A *traffic manager* identifies appropriate global traffic management strategies.

The proposed model has been implemented and validated in MATISSE 2.0 [4], a large-scale multi-agent based traffic simulation platform. Experimental results show that the proposed model improves the evacuation effort. In addition, the evacuation operation is able to cope quickly and effectively with dynamic road closures.

This paper is organized as follows: Sect. 2 discusses related works in urban traffic evacuation. Section 3 gives an overview of ATS. Section 4 presents a re-organizing traffic network for urban evacuations, and Sect. 5 provides an analysis of the experimental results.

2 Related Work

Several studies have been conducted to investigate various traffic control strategies for urban evacuations. Among these strategies, road reversal operations have been considered by traffic engineers and researchers as one of the most effective methods to enhance traffic flow during evacuations [11,22,24,25]. Road reversal operations refer to the process of reversing the direction of one or more travel lanes of an inbound traffic movement to be used for the movement of traffic at the opposing outbound direction [5]. As in other evacuation strategies, the planning of road reversal operations have been investigated for application *before* and *after* the occurrence of a disaster.

During the *before-disaster road reversal planning*, traffic engineers specify various road reversal configurations with respect to criteria such as evacuation time or formation of bottlenecks [13,24]. Microscopic traffic simulations are used to assess the effectiveness of the predefined configurations. This conventional approach does not provide a systematic methodology to find an optimal road reversal configuration. Instead, road reversals are planned according to scenario analysis and the use of historical data. Studies have shown that, predefining road reversal configurations may be inadequate for unexpected disaster situations.

The *after-disaster road reversal planning* involves the systematic generation of road reversal settings to support the evacuation efforts, emergency relief services, and the recovery of the traffic network function. According to the taxonomy in [12], the after-disaster road reversal planning can be classified as *response* and *recovery operations*.

The *response* operations refer to the road reversals applied immediately after the disaster to enhance the evacuation throughput and/or support emergency services. The existing response operations are based on: (1) the advanced identification of the optimal road reversal settings or (2) the application of dynamic road reversal operations according to the instantaneous traffic conditions. In the advanced identification of road reversal settings, the execution of mathematical models is used to determine upfront the optimal road-reversal settings for the entire evacuation process. To this effect, a variety of AI techniques are used [16,26,28]. The general approach follows the *iterative optimization-evaluation model*: at each iteration, the optimization process applies an intelligent heuristic to find a set of effective road reversal configurations. Results show that the execution of these heuristics is time-consuming and unfit to adapt to unforeseen conditions that commonly occur in highly dynamic traffic environments [9]. In addition, road closures caused by accidents or road damages might compromise the generated road reversal configuration even before it is implemented. In addition, the road reversal configurations generated by these techniques may result in a rigid traffic network topology that can no longer be dynamically re-organized. An alternative lies in seeking an adaptive solution that can continuously adjust to the dynamic traffic conditions.

Several studies have investigated the use of fully autonomous vehicles that can adapt in real-time to the dynamic change in road reversal configurations [7,10]. In these models, optimal road reversal configurations are dynamically generated and implemented according to the immediate traffic conditions. In spite of the theoretical effectiveness of these models, autonomous vehicles are still not widely used and there is a need to investigate dynamic road reversal techniques that consider human drivers and make use of the infrastructure only.

The after-disaster road reversals have been also applied to support *recovery* operations [1,14,18]. Similarly to response road-reversal planning, the generated plans are based on the execution of heavy mathematical models that do not take into consideration dynamic traffic conditions.

In the traffic literature, there is an increasing interest in studying *resilience* [17,19], but, unfortunately, the amount of published research is still limited. Until now, resilience has been discussed from perspectives including [15]: the definition of properties (e.g., readiness, response, recovery) and metrics; the definition of organizational and legal structures to implement it; the development of conceptual models for its analysis. None of the existing works published in the literature have investigated the ability to recover the evacuation operation after the application of a road reversal strategy.

In this paper, we present a resilient re-organizing traffic environment for urban evacuations in an agent-based ITS. Our proposed model is implemented in MATISSE 2.0, a large-scale multi-agent based traffic simulation platform. The re-organizing model improves on the state of the art in re-organizing traffic networks for evacuation as follows: (1) it implements road reversal operations that consider the traffic conditions to implement a safe road reversal; (2) it maintains a resilient network topology that is re-organized dynamically to recover its evacuation functionality following road closures; (3) it is based on a combination of road reversal and zoning strategies; (4) it does not consider autonomous vehicles but only infrastructure-to-infrastructure communications in the context of ATS.

3 A Model for an Agent-Based ITS

In this section we discuss ATS, a model for an agent-based intelligent transportation system. As shown in Fig. 1, ATS consists of various types of agents, each having individual concerns [27].

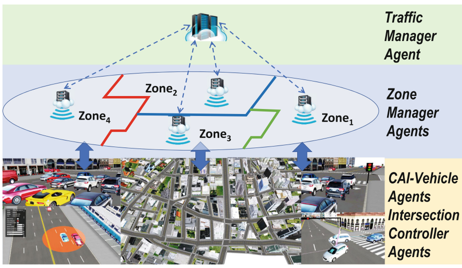


Fig. 1. ATS infrastructure

Zone Manager Agent: In ATS, the traffic environment space is partitioned into areas called *zones*. A zone is assigned a specialized agent called *zone manager*. A zone manager is responsible for maintaining a resilient network topology within its zone. To this effect it: (1) analyzes information about the traffic flow at each intersection within its zone to detect road closures; (2) defines road reversals on major roads within its zone to construct evacuation routes; (3) diverts the traffic to alternative evacuation routes in case of road closures; and (4) informs vehicles within its zone about the changed network topology.

Traffic Manager Agent: In addition to zone managers, a hierarchy of meta-level agents called *traffic managers* identify appropriate global traffic management strategies. In this paper, we restrict our discussion to one traffic manager. The traffic manager is responsible for: (1) defining the initial traffic zones; (2) communicating with zone managers to obtain information about road closures that cannot be recovered locally within their zones; (3) re-organizing the zones' boundaries to rapidly regain the evacuation functionality.

Intersection Controller Agent: In ATS, each intersection is assigned an *intersection controller agent* whose role is to control the traffic flow on roads. A controller agent manages digital traffic devices which are intended to replace our current traffic lights and sign panels. The intersection controller agent is responsible for interacting with its zone manager to: (1) implement a road reversal operation; (2) divert the traffic to specific routes; (3) provide its zone manager with the number of incoming and outgoing vehicles at its intersection.

For instance, to reverse the direction of a road, the intersection controller agents at the source and target intersections change their display settings to define the new traffic streams driving through the reversed road and to ensure non-conflicting traffic streams at their intersections. Detailed information on the implementation of road reversals can be found in [2].

Context-Aware Intelligent (CAI) Vehicle Agent: A CAI-vehicle is a vehicle equipped with an agent-based system which: (1) monitors the driver's behavior and the environment, (2) communicates with other vehicles, (3) communicates with intersection controllers in case of immediate emergency, and (4) communicates traffic information/routing to the vehicle's driver.

The traffic network re-organization model discussed in this paper builds on the short results discussed in [3]. We introduce the concept of resilience and improve the agent algorithms to allow for network connectivity, travel time and network capacity. Detailed discussions of the algorithms are given in the next section.

4 Re-organization Algorithms

In this section we provide a detailed discussion about the algorithms used in our re-organizing model.

4.1 General Overview

In order to better utilize the available network capacity and to distribute the evacuated traffic efficiently over the exits, the traffic manager partitions the traffic network into small areas called *zones*. The number of zones depends on the number of evacuation exits in the traffic environment. A zone is defined for each evacuation exit and is assigned a zone manager agent.

At initialization time, the zone boundaries are defined to achieve an evenly distributed traffic across zones in proportion to the zones' capacities. A zone $\mathcal{Z}_x$'s capacity $\mathcal{C}_{\mathcal{Z}_x}$ is defined as the maximum number of vehicles that can be accommodated within the zone's roads. As such, $\mathcal{C}_{\mathcal{Z}_x}$ can be defined as: $\mathcal{C}_{\mathcal{Z}_x} = \sum_{r \in \mathcal{Z}_x.roads}(\frac{r.length \times r.lanes}{v.length})$, where $\mathcal{Z}_x.roads$ is the set of roads in $\mathcal{Z}_x$; $r.length$ is the length of the road r; $r.lanes$ is the number of lanes in r; and $v.length$ is the average vehicle length.

To define the zones, the traffic manager iterates over the created zones and incrementally adds roads to these zones starting from the closest roads to exits. At each iteration, the traffic manager observes the traffic distribution across zones and decides the next set of roads to be added to each zone that maintains a balanced traffic densities. A traffic density for $(\mathcal{Z}_x)$ is defined as: $\mathcal{Z}_x.\textsc{Density} = \frac{\mathcal{Z}_x.\mathcal{V}}{\mathcal{C}_{\mathcal{Z}_x}}$, where $\mathcal{Z}_x.\mathcal{V}$ is the number of vehicles in $\mathcal{Z}_x$. Figure 2 shows an example of a traffic environment partitioned into three zones around the exits.

In the proposed model, re-organization of the traffic environment occurs within and across zones:

1. Re-organization Within the Traffic Zone: Each zone manager defines a road reversal plan for major evacuation routes within its zone. The purpose of this plan is to maximize the traffic flow towards its exit. To this effect, the zone manager implements road reversals on evacuation routes that connect each of its border intersections to its evacuation exit. In addition, the road reversal configuration is specified to achieve the best possible traffic network resilience that allows to recover the evacuation functionality after severe road damages.

During evacuation times, the zone manager collaborates with its intersection controller agents to detect road closures. Once a road closure is detected, the zone manager directs the respective intersection agents to divert the blocked traffic to alternative evacuation routes. In case the recovery of the evacuation functionality is not attainable locally within the zone, the zone manager requests from the traffic manager to divert the traffic to the neighboring zones.

2. Re-organization Across Traffic Zones: If any of the zone managers are no longer able to evacuate part or all of its traffic due to severe road damages, the traffic manager dynamically reorganizes the zones' boundaries to rapidly evacuate the maximum number of blocked vehicles.

Re-organizing the zones includes transferring roads and intersections across zones. As such, the affected zone managers re-organize the traffic network within theirs zones to better utilize the capacity of the newly added roads. To this end, the zone manager defines a road reversal configuration on a subset of the newly received roads to maximize the road capacity towards its exit while maintaining a resilient traffic network.

4.2 Re-organization Within the Traffic Zone

Each zone manager defines and implements a road reversal plan for major evacuation routes within its zone. The purpose of this plan is to maximize the traffic flow towards its evacuation exit. This is achieved by defining evacuation routes that connect each of its border intersections to its evacuation exit. The detailed plan is given in Algorithm 1.

The zone manager starts by finding the set of shortest paths (ℓ) that connect each of the $\mathcal{Z}_x$'s border nodes to its evacuation exit Σ (see Algorithm 1 Section(A)). Figure 2 shows an example of a three-zone environment with shortest paths defined from each of the zone's border nodes to the zone's evacuation exits.

Algorithm 1. Constructing Evacuation Routes within ($\mathcal{Z}_x$)

```
begin
    /* *********** A *********** */
    foreach b ∈ BorderNodes(Z_x) do
        sp_b ← FindShortestPath(b, Σ)
        ℓ.add(sp_b)
    if ¬Resilient(Z_x, ℓ) then
        /* *********** B *********** */
        χ ← arg max_{b∈B} Distance(b, Σ)
        foreach sp_b ∈ ℓ do
            foreach Z_y ∈ Zones|Z_y ≠ Z_x do
                sp_b.S_{Z_y} ← 0
                foreach n ∈ Z_x.nodes do
                    if IsReachable(n, z, ¬sp_b) then
                        sp_b.S_{Z_y} += χ − Distance(n, Σ) + 1
            /* *********** C *********** */
            sp_b.S_all ← Σ_{Z_y∈Zones|Z_y≠Z_x} (sp_b.S_{Z_y})
        /* *********** D *********** */
        sp_r ← arg max_{sp_b∈ℓ} sp_b.S_all
        Ω.add(sp_r)
        ℓ.remove(Ω)
        ReverseRoads(ℓ)
```

The zone manager proceeds to evaluate the resilience of $\mathcal{Z}_x$ in case all roads in ℓ are considered for reversal. To this effect, the zone manager checks for the existence of a path between each intersection attached to Σ and any neighboring zone. In case $\mathcal{Z}_x$ is not resilient, the zone manager executes Algorithm 1 (Section B) to eliminate the minimum set of shortest paths (Ω) from ℓ in order to improve resilience. This consists of finding the road reversal

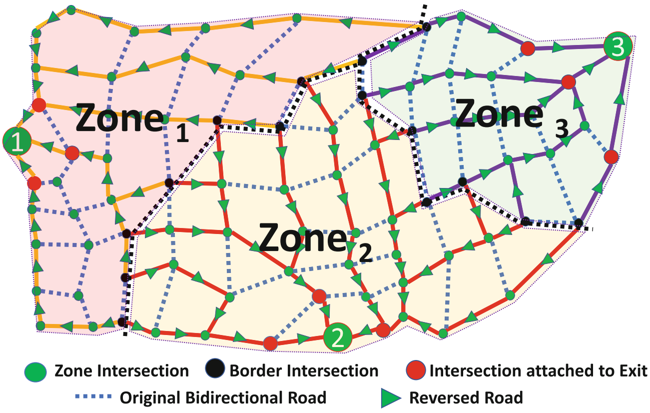


Fig. 2. Constructing reversal paths within zones

configuration that makes the maximum number of intersections in $\mathcal{Z}_x$ connected with paths to the maximum number of neighboring zones.

The process starts by computing Ω, the minimum subset of shortest paths that must be removed from ℓ to improve resilience. To this effect, the zone manager iterates over the set of shortest paths in ℓ and for each neighboring zone $\mathcal{Z}_y$, it computes a score $(sp_b.\mathcal{S}_{\mathcal{Z}_y})$ that represents the degree of resilience to divert the traffic from $\mathcal{Z}_x$ to $\mathcal{Z}_y$ (see Algorithm 1 (Section B)). As the shortest path is defined from a border intersection to the intersection attached to Σ, all shortest paths sharing the same intersection attached to Σ are eliminated together.

The score $sp_b.\mathcal{S}_{\mathcal{Z}_y}$ is calculated by summing the scores gained by each intersection in $\mathcal{Z}_x$ with a path to $\mathcal{Z}_y$. Intersections closer to the exit are assigned higher scores. Hence, the score gained by each intersection is defined as: $(\chi - \text{DISTANCE}(n, \Sigma) + 1)$, where χ is the maximum distance between any border node in $\mathcal{Z}_x$ and Σ and $\text{DISTANCE}(n, \Sigma)$ is the distance between n and Σ. We choose to measure the distance in terms of the number of intersections as they are considered as the main source of travel delays [6,26].

Then, the zone manager computes the overall resilience score $sp_b.\mathcal{S}_{all}$ by summing all $sp_b.\mathcal{S}_{\mathcal{Z}_y}$ obtained for each of the neighboring zones (see Algorithm 1 (Section 1)). After that, the zone manager adds the shortest path with the highest overall resilience score $sp_b.\mathcal{S}_{all}$ to Ω.

Finally, the zone manager removes Ω from ℓ and directs its intersection controller agents to initiate the implementation of road reversals at ℓ (see Algorithm 1 (Section D)).

For instance, in the scenario depicted in Fig. 3. The number beside each intersection represents its distance from Σ (the green circle numbered 2). In the illustrated example, $Zone_2$'s manager has multiple options to eliminate a defined shortest path Ω (due to limited space, we illustrate the best three options).

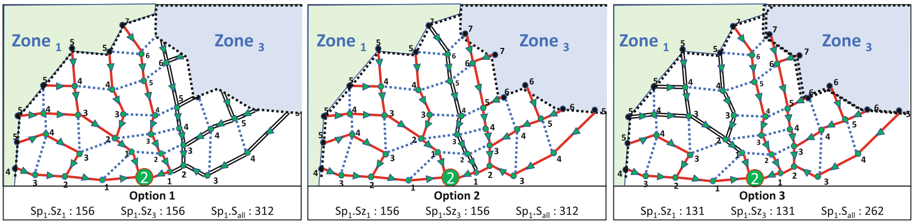


Fig. 3. Evaluating resilience in $Zone_2$ (Color figure online)

As can be seen, option 1 and 2 achieve the highest resilience scores for the neighboring $Zone_1$ and $Zone_3$.

Special Cases. In some cases, the use of resilience scores alone is not enough to decide Ω. Next, we discuss two special cases that requires a further evaluation from the zone manger.

1. **Equal overall resilience scores:** In case there are two or more shortest paths with equal $sp_b.s_{all}$ scores, the zone manager chooses the shortest path with the minimum travel distance ($sp_b.\mathcal{D}$). The travel distance ($sp_b.\mathcal{D}$) is computed by summing the shortest distances between each intersection in $\mathcal{Z}_x$ and its nearest reachable neighboring zone (travel time resilience). In case the value of $sp_b.\mathcal{D}$ for two or more shortest paths are equal, the traffic manager chooses the shortest path that has the minimum road capacity (capacity resilience).
2. **The eliminated shortest path is not enough:** In some cases, more than one shortest path need to be eliminated to make a neighboring zone $\mathcal{Z}_y$ reachable by a reasonable number of intersections. By reasonable, we mean that the percentage of intersections with a path to any $\mathcal{Z}_y$ must exceed a threshold ρ. To handle this case, the zone manager searches for two or more shortest paths to be eliminated together to reach the required ρ and considers them as a unified sp when executing Algorithm 1. Although this step might result in a large waste of the additional capacity gained by road reversals, it is inevitable to achieve resilience.

Evacuation Recovery Within the Zone: Although implementing road reversals enhances the traffic flow towards the exit, a road closure might block the traffic from reaching the exit. Therefore, intersection control agents continuously monitor the traffic flow on evacuation routes and send this information to their zone managers. The zone manager analyses the received information to detect closures. Once a closure is detected on a road (κ), the zone manager implements a recovery plan to divert the traffic incoming to κ to alternative evacuation routes within its zone.

Figure 4 illustrates a scenario where a road attached to the evacuation exit in $Zone_2$ (marked with X) is fully closed. In this case, the traffic flowing to the exit through the closed road is blocked. Therefore, the zone manager diverts the

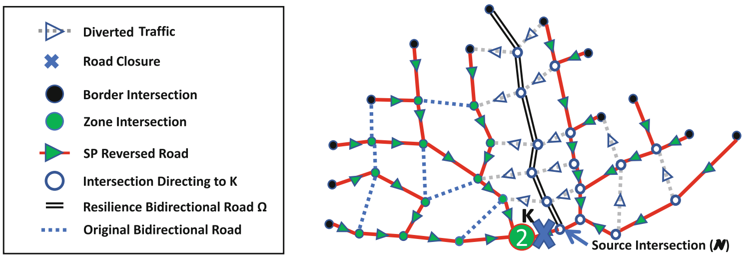


Fig. 4. Recovery within the traffic zone

traffic flow at each intersection in the blocked routes (marked with empty circle) to its neighbor intersection with the shortest distance to the exit.

4.3 Re-organization Across Traffic Zones

Upon receipt of the assistance request from a zone manager, the traffic manager executes Algorithm 2 to reorganize the zones and effectively evacuate the maximum number of blocked vehicles.

Algorithm 2. Re-Organization accorss Zones

input : $\mathcal{I}$: List[1..n] of Intersections
begin
  $\mathcal{Z} \leftarrow$ ReachableZones($\mathcal{I}$)
  if $\mathcal{Z}.count = 1$ **then**
    `/* ************* A ************* */`
    TransferIntersections($\mathcal{I}, \mathcal{Z}$)
  else if $\mathcal{Z}.count \geq 1$ **then**
    `/* ************* B ************* */`
    $\Pi \leftarrow$ Split($\mathcal{I}, \Omega$)
    foreach $\pi \in \Pi$ **do**
      $\mathcal{Z}_{closest} \leftarrow arg\,min_{z \in \mathcal{Z}} \sum_{i \in \pi}$ Distance(i, Σ_z)
      Transfer($\pi, \mathcal{Z}_{closest}$)
    `/* ************* C ************* */`
    foreach $sp \in \Omega$ **do**
      $\mathcal{Z}_{closest} \leftarrow arg\,min_{z \in \mathcal{Z}} \sum_{i \in sp}$ Distance(i, Σ_z)
      TransferSafe($sp.nodes, \mathcal{Z}_{closest}$)

The process starts by finding the neighboring zones that can be reached by the blocked traffic in $\mathcal{I}$. In case the traffic in $\mathcal{I}$ can only reach one neighboring zone, the traffic manager requests from the respective zone manager to transfer the set of intersections in $\mathcal{I}$ to its neighboring zone (see Algorithm 2 (Section A)).

Figure 5 shows an example of isolated intersections in $Zone_2$ that can only reach $Zone_3$. Hence, the four isolated intersections are transferred to $Zone_3$.

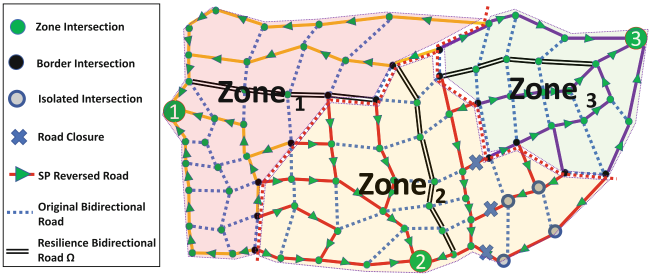


Fig. 5. Isolated intersections - one reachable zone

Otherwise, if the traffic in $\mathcal{I}$ can reach multiple neighboring zones, the traffic manager splits $\mathcal{I}$ among the reachable zones (see Algorithm 2 (Section B)). To ensure that the transferal of intersections among zones does not result in deadlocks for the blocked traffic in $\mathcal{I}$, the traffic manager splits the isolated intersections into partitions separated by the resilience routes in Ω. Then, it transfers each split (π) to its closest reachable zone. A closest reachable zone is defined as the zone that has the minimum sum of shortest distances between each intersection in π and its exit. For instance, in the scenario illustrated in Fig. 6(a), all roads attached to the exit in $Zone_2$ are closed. Consequently, all intersections in $Zone_2$ become isolated. As both $Zone_1$ and $Zone_3$ are reachable to the isolated intersections, the traffic manager splits the isolated intersections based on the resilience route marked with double black lines to π_1 and π_3. As can be obviously seen, π_1 and π_3 are transferred to $Zone_1$ and $Zone_3$ respectively based on their distance from each zone (see Fig. 6(b)).

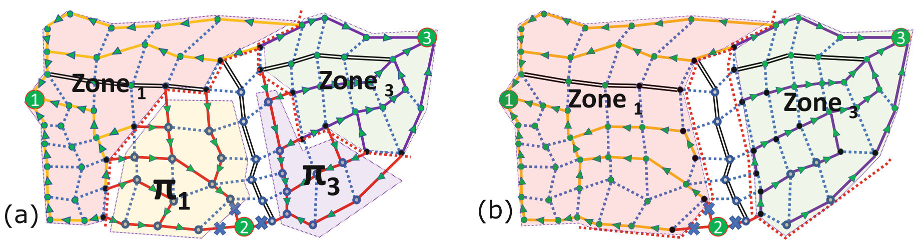


Fig. 6. Isolated intersections - multiple reachable zones

Once the zone manager receives the newly transferred intersections, it reorganizes the roads attached to these intersections according to the approach described in Algorithm 1. For instance, Fig. 6(b) illustrates the result of transferring π_1 and π_3 to $Zone_1$ and $Zone_3$. As can be seen, there is an evacuation route from each new border intersection to its zone's evacuation exit. Other roads in the zone are set as bidirectional. Furthermore, the bidirectional resilience route at $Zone_2$ is not transferred until traffic diversion from $Zone_2$ to the new zones is completed. Upon completion, the resilience road in $Zone_2$ is transferred to its closest reachable zone ($Zone_3$).

5 Experimental Results

In this section, we start by discussing the model implementation and the settings used to run the experiments. Then, we show how the proposed environment outperforms other environments in terms of resilience and evacuation throughput.

5.1 Experiment Settings

The proposed re-organizing model has been implemented and tested using MATISSE 2.0 [4], a large-scale multi-agent based traffic simulation platform. In MATISSE, traffic elements are modeled as virtual agents equipped with perception and communication mechanisms that allow them to perceive their surroundings and interact with each other.

The experiment takes place in a virtual traffic network that consists of 145 roads and 83 intersections. To ensure a fair assessment, we fix the number of lanes to 3. Also, four evacuation exits are specified at the outskirt of the traffic network. Vehicles are removed from the simulation once they reach these exits. Initially, vehicles drive randomly in the traffic network when an emergency siren is triggered. Each vehicle attempts to reach an evacuation exit. After 1000 simulation cycles, two explosions are triggered near exits 2 and 4 making them unreachable.

To evaluate the effectiveness of the proposed model, we run the same evacuation scenario using six different traffic models obtained by combining different types of road reversal and zoning strategies. The use of these combinations aims to show the effect of the various road reversal and zoning strategies on resilience and evacuation throughput.

Flexible Zones - No Reversal (Flexible-NO)**:** this model contains four zones which do not implement road reversal strategies. The zones' boundaries can be redefined by the traffic manager.

One Zone - All Reversal (1Z-ALL)**:** this model contains only one zone. Road reversals are uniformly applied for the shortest paths between all intersections and their nearest exits.

Fixed Zones - All Reversal (Fixed-ALL)**:** this model contains four equally sized zones. For each zone, road reversals are uniformly applied for the shortest paths between all intersections and their nearest exit.

Flexible Zones - All Reversal (Flexible-ALL)**:** this model contains four zones. For each zone, road reversals are uniformly applied for the shortest paths between

all intersections and their nearest exit. The zones' boundaries can be redefined by the traffic manager.
Fixed Zones - SP Reversal (Fixed-SP)**:** this model contains four equally sized zones. For each zone, road reversals are applied for the shortest paths between all intersections in the zone and their zone's exit.
Flexible Zones - SP Reversal (Flexible-SP)**:** this model refers to the model discussed in Sect. 4. It contains four zones. For each zone, road reversals are applied for the shortest paths between all intersections in the zone and their zone's exit. The zones' boundaries can be redefined by the traffic manager.

For each of the aforementioned models, we run the evacuation scenario 10 times and with 3500 vehicles. Using these runs, we compute: (1) the average accumulated number of evacuated vehicles over time and (2) the average number of reversed roads. The simulation cycle time of 0 refers to the start of the evacuation process. Threshold ρ is set to 0.75. The simulation scenarios were executed on a single multicore PC (Intel Core i7 X980 CPU (3.33 GHz), 8.00 GB) [1].

5.2 Results and Evaluation

To better understand the effect of these models on resilience, we discuss the evacuation performance *before* and *after* the occurrence of road closures.

Evacuation Performance Before Road Closures: Figure 7 shows the accumulated number of evacuees over time. Initially, all traffic models have a comparable number of accumulated evacuees. However, as time elapses, the models implementing road reversals outperform the other models in terms of the number of accumulated evacuees. This is to be expected as the additional road capacity gained by road reversals improves the traffic flow towards the exits. For instance, Flexible-NO has around 900 evacuees at simulation cycle time 1000 and (1Z-ALL) has around 1400 evacuees at that time.

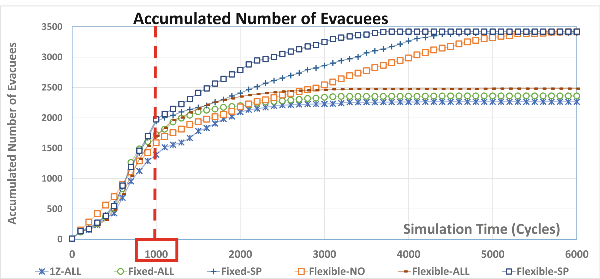


Fig. 7. Accumulated number of evacuees

[1] Demonstrations for the experiments are available at http://www.utdmavs.org/its/.

Despite the fact that road reversals improve the evacuation throughput rate, the results show that the accumulated number of evacuees using the Flexible-NO initially exceeds all other models. This is explained by the temporary negative impact caused by road reversals. During the reversal setup time, the new traffic incoming to roads undergoing reversal is forced to flow to alternative roads which become denser. When the benefits gained by road reversals surpass their initial set-up effect, Flexible-SP outperforms the other models. This starts at simulation cycle time of 600 in Fig. 7.

The results given in Fig. 7 also show that the evacuation throughput of Flexible-SP increases asymptotically and outperforms the other models. For instance, Flexible-SP evacuates the largest number of vehicles (1975) compared to the other models by simulation cycle time 1000. This leads us to conclude the presence of the flexible zoning strategy results in an improved utilization of the additional network capacity gained by road reversals.

Evacuation Performance After Road Closures: As indicated in Fig. 7, Flexible-SP is able to adapt faster to the closure of $Zone_2$'s and $Zone_4$'s exits and clearly outperforms the other models. For instance, Flexible-SP completes the evacuation of its traffic by simulation cycle time 3700 while Fixed-SP completes the evacuation by the simulation cycle time of 4700. As discussed in Sect. 4, in case of traffic diversion across zones, Flexible-SP partitions the blocked traffic into splits and transfers each split to its nearest neighboring zone. Hence, the use of a flexible zoning strategy improves the travel time resilience.

In addition, when using a flexible zoning strategy, as the traffic flows towards the exit, the roads far from the exit become almost empty and are quickly reversed. This is illustrated in Fig. 8 which shows the accumulated number of implemented road reversals. As shown in this figure, after simulation cycle time 1000, the number of implemented road reversals increases asymptotically. As such, the diverted traffic exploits the additional capacity at these routes to reach the new zones and evacuate. Moreover, $Zone_1$ and $Zone_3$ have low traffic densities by the time the diverted traffic crosses their boundaries. Consequently, this improves the capacity resilience of the network and the diverted traffic rapidly evacuates as in the case of a staged evacuation.

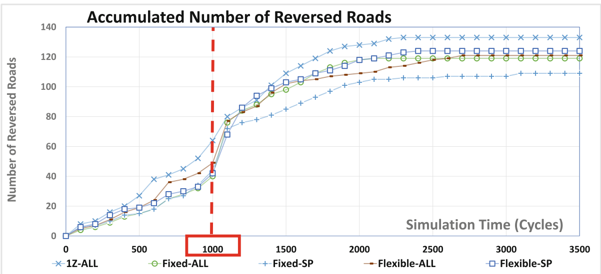


Fig. 8. Accumulated number of reversed roads

The results given in Fig. 7 show that the models implementing ALL road reversals are not able to cope with closures of $Zone_2$ and $Zone_4$'s exits and fail to evacuate all of their traffic. For example, (1Z-ALL), (Fixed-ALL), and (Flexible-ALL) evacuate 2265, 2361, and 2481 vehicles respectively. This is due to the rigid traffic network that results from the excessive implementation of road reversals which prevents the blocked traffic from reaching other neighboring zones. This leads us to conclude that traffic models incorporating road reversals without considering resilience are not reliable in case of unexpected perturbations.

6 Conclusion

In this paper we discussed a model for a resilient agent-based re-organizing traffic network for urban evacuations. The proposed model aims to improve the evacuation efficiency after severe and unexpected disruptions. To achieve this goal, the proposed model integrates the use of road reversal with zoning strategies. The experimental results show that the proposed model effectively improves the evacuation throughput.

The results also show that road reversal operations consume a considerable amount of time which negatively impacts the effectiveness of traffic evacuation. Our short term goal is to investigate improved dynamic reversal strategies that will make use of the cleared road capacity during reversal operations. We also plan to investigate the use of *crossing elimination* and *adaptive signal timings.*

References

1. Aksu, D.T., Ozdamar, L.: A mathematical model for post-disaster road restoration: enabling accessibility and evacuation. Transp. Res. Part E: Logistics Transp. Rev. **61**(Suppl. C), 56–67 (2014)
2. Al-Zinati, M., Wenkstern, R.: Simulation of traffic network re-organization operations. In: Proceedings of the 20th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications, pp. 178–186, September 2016
3. Al-Zinati, M., Wenkstern, R.Z.: An agent-based self-organizing traffic environment for urban evacuations. In: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, Sao Paulo, Brazil, pp. 1454–1456 (2017)
4. Al-Zinati, M., Zalila-Wenkstern, R.: Matisse 2.0: a large-scale multi-agent simulation system for agent-based ITS. In: 2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Singapore, pp. 328–335, December 2015
5. American Association of State Highway and Transportation Officials: Policy on Geometric Design of Highways and Streets. Sixth edn (2011)
6. Cova, T.J., Johnson, J.P.: A network flow model for lane-based evacuation routing. Transp. Res. Part A: Policy Practice **37**(7), 579–604 (2003)
7. Duell, M., Levin, M.W., Boyles, S.D., Waller, S.T.: System optimal dynamic lane reversal for autonomous vehicles. In: 18th International Conference on Intelligent Transportation Systems, Las Palmas, Spain, pp. 1825–1830, September 2015

8. Faturechi, R., Miller-Hooks, E.: Measuring the performance of transportation infrastructure systems in disasters: a comprehensive review. J. Infrastruct. Syst. **21**(1), 04014025 (2015)
9. Gershenson, C.: Self-organizing urban transportation systems. In: Portugali, J., Meyer, H., Stolk, E., Tan, E. (eds.) Complexity Theories of Cities Have Come of Age, pp. 269–279. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-24544-2_15
10. Hausknecht, M., Au, T.C., Stone, P., Fajardo, D., Waller, T.: Dynamic lane reversal in traffic management. In: 14th International Conference on Intelligent Transportation Systems (ITSC), Washington DC, USA, pp. 1929–1934, October 2011
11. Kim, S., Shekhar, S., Min, M.: Contraflow transportation network reconfiguration for evacuation route planning. IEEE Trans. Knowl. Data Eng. **20**(8), 1115–1129 (2008)
12. Konstantinidou, M., Kepaptsoglou, K., Karlaftis, M.: Transportation network post-disaster planning and management: a review part I. Int. J. Transp. **2**(3), 1–16 (2014)
13. Lim, E., Wolshon, B.: Modeling and performance assessment of contraflow evacuation termination points. Transp. Res. Rec.: J. Transp. Res. Board **1922**(1), 118–128 (2005)
14. Matisziw, T.C., Murray, A.T., Grubesic, T.H.: Strategic network restoration. Netw. Spatial Econ. **10**(3), 345–361 (2010)
15. Mattsson, L.G., Jenelius, E.: Vulnerability and resilience of transport systems - a discussion of recent research. Transp. Res. Part A: Policy Practice **81**(Suppl. C), 16–34 (2015)
16. Meng, Q., Khoo, H.L., Cheu, R.L.: Microscopic traffic simulation model-based optimization approach for the contraflow lane configuration problem. J. Transp. Eng. **134**(1), 41–49 (2008)
17. Miller-Hooks, E., Zhang, X., Faturechi, R.: Measuring and maximizing resilience of freight transportation networks. Comput. Oper. Res. **39**(7), 1633–1643 (2012)
18. Ozdamar, L., Aksu, D.T., Ergüneş, B.: Coordinating debris cleanup operations in post disaster road networks. Socio-Econ. Plann. **48**(4), 249–262 (2014)
19. Snelder, M., van Zuylen, H., Immers, L.: A framework for robustness analysis of road networks for short term variations in supply. Transp. Res. Part A: Policy Practice **46**(5), 828–842 (2012)
20. Tuydes, H., Ziliaskopoulos, A.: Tabu-based heuristic approach for optimization of network evacuation contraflow. Transp. Res. Rec.: J. Transp. Res. Board **1964**(1), 157–168 (2006)
21. Wang, J.W., Wang, H.F., Zhou, Y.M., Wang, Y., Zhang, W.J.: On an integrated approach to resilient transportation systems in emergency situations. Nat. Comput. 1–9 (2017)
22. Wang, J., Ip, W., Zhang, W.: An integrated road construction and resource planning approach to the evacuation of victims from single source to multiple destinations. IEEE Trans. Intell. Transp. Syst. **11**(2), 277–289 (2010)
23. Wang, J., Wang, H., Zhang, W., Ip, W., Furuta, K.: Evacuation planning based on the contraflow technique with consideration of evacuation priorities and traffic setup time. IEEE Trans. Intell. Transp. Syst. **14**(1), 480–485 (2013)
24. Wolshon, B.: "ONE-WAY-OUT": contraflow freeway operation for hurricane evacuation. Nat. Hazards Rev. **2**(3), 105–112 (2001)
25. Wolshon, B., Urbina Hamilton, E., Levitan, M., Wilmot, C.: Review of policies and practices for hurricane evacuation. II: traffic operations, management, and control. Nat. Hazards Rev. **6**(3), 143–161 (2005)

26. Xie, C., Turnquist, M.A.: Lane-based evacuation network optimization: an integrated Lagrangian relaxation and tabu search approach. Transp. Res. Part C: Emerg. Technol. **19**(1), 40–63 (2011)
27. Zalila-Wenkstern, R., Steel, T., Leask, G.: A self-organizing architecture for traffic management. In: Weyns, D., Malek, S., de Lemos, R., Andersson, J. (eds.) SOAR 2009. LNCS, vol. 6090, pp. 230–250. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14412-7_11
28. Zhao, X., Ren, G., Huang, Z.: Optimizing one-way traffic network reconfiguration and lane-based non-diversion routing for evacuation. J. Adv. Transp. **50**(4), 589–607 (2016)

Coping with Bad Agent Interaction Protocols When Monitoring Partially Observable Multiagent Systems

Davide Ancona, Angelo Ferrando, Luca Franceschini, and Viviana Mascardi(✉)

Università degli Studi di Genova, Genoa, Italy
{davide.ancona,angelo.ferrando,luca.franceschini,
viviana.mascardi}@dibris.unige.it

Abstract. Interaction Protocols are fundamental elements to provide the entities in a system, be them actors, agents, services, or other communicating pieces of software, a means to agree on a global interaction pattern and to be sure that all the other entities in the system adhere to it as well. These "global interaction patterns" may serve different purposes: if the system does not yet exist, they may specify the allowed interactions in order to drive the system's implementation and execution. If the system exists before and independently from the protocol, the protocol may still specify the allowed interactions, but it cannot be used to implement them. Its purpose in this case is to monitor that the actual system does respect the rules (runtime verification). Tagging some protocols as good ones and others as bad is common to all the research communities where interaction is crucial, and it is not surprising that some protocol features are recognized as bad ones everywhere. In this paper we analyze the notion of good, bad and ugly protocols in the MAS community and outside, and we discuss the role that bad protocols, despite being bad, may play in a runtime verification scenario where not all the events and interaction channels can be observed.

Keywords: Bad agent interaction protocols · Monitoring
Distributed Runtime Verification · Partial observability

1 Introduction

Interaction Protocols are a key ingredient in MASs as they explicitly represent the agents expected/allowed communicative patterns and can be used either to check the compliance of the agent actual behavior w.r.t. expected one [1,10] or to drive the agent behavior itself [4].

Interaction protocols are also crucial outside the MAS community: what we name an "Agent Interaction Protocol", AIP, is referenced as a "Choreography" in the Service Oriented Computing (SOC) community [36] and as a "Global Type" in the multiparty session types one [15,31]. In the MAS community, AIPs describe interaction patterns characterizing the system as a whole. This global

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 59–71, 2018.
https://doi.org/10.1007/978-3-319-94580-4_5

viewpoint is supported by many formalisms and notations such as AUML [32], commitment machines and their extensions [11,12,22,40,42], the Blindingly Simple Protocol Language (BSPL) and its Splee extension [21,37], the Hierarchical Agent Protocol Notation (HAPN) [41], trace expressions [7]. A systematic comparison of these approaches – apart from AUML which is not formal, and not supported by a textual notation – can be found in [9].

When moving from the specification to the execution stage, the AIP must be enacted by agents in the MAS: besides the global description of the protocol, the "local" description of the AIP portion each agent is in charge of, is required to run the AIP. The AIP enactment is usually left to Computer-Aided Software Engineering tools that move from AIP diagrams directly into agent skeletons in some concrete agent oriented programming language [23,30], or to algorithms that translate the AIP textual representation to some abstract, intermediate formalism for modeling the local viewpoint [19,26]. Such intermediate formalisms are not perceived as the main target of the MAS research and no standardization effort has been put on them. In the SOC community, on the contrary, formalisms exist for modeling both the global and the local perspectives. As observed by [35], WS-CDL[1] follows an interaction-oriented ("global") approach, whereas in BPEL4Chor[2] the business process of each partner involved in a choreography is specified using an abstract version of BPEL[3]: BPEL4Chor follows a process-oriented ("local") approach. In the multiparty session types community, the main emphasis is on type-checking aspects: the formalism used to represent global types is relevant, as well as its expressive power, but even more relevant are the properties of the "global to local" projection function w.r.t. type-safety [25]. Finally, in the Distributed Runtime Verification (DRV) community [29], one of the most relevant issues is how to automatically generate a monitor (or a set of monitors) from a given protocol, and to use it (them) to dynamically monitor the existing system's behaviour.

Our work lies in the intersection of the MAS and DRV research areas. We are interested in exploring under which conditions a good AIP can turn into a bad one, if it is possible to exploit such bad protocols anyway for DRV purposes, and in providing an implemented tool for automatically generating a set of MAS monitors under these conditions. To this aim, we developed RIVERtools [27] which supports the development of AIPs modeled using the trace expressions formalism [3,7] via a user-friendly GUI. RIVERtools implements static controls to check if a protocol is good, ugly or bad, and supports the user in finding out how to (partially) decentralize the runtime verification process also in the last case [28]. The adoption of RIVERtools for this purpose in exemplified in the companion demo paper [6], which addresses the most practical aspects of our investigation: this paper is mainly centered around the notions of bad, good, and

[1] Web Services Choreography Description Language Version 1.0 W3C Candidate Recommendation 9 November 2005, https://www.w3.org/TR/ws-cdl-10/.

[2] BPEL4Chor Choreography Extension for BPEL, http://www.bpel4chor.org/.

[3] Web Services Business Process Execution Language Version 2.0, OASIS Standard, 11 April 2007, http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html.

ugly protocol in the literature, and on the observability-driven transformation of AIPs which, under known environmental/network constraints, may turn a good protocol into a bad one which can nevertheless serve for DRV purposes.

2 The Good, the Bad and the Ugly

The Good. Let us consider the following interaction protocol expressed in natural language:

1. Alice sends a whatsApp message to her mother Barbara asking her to buy a book (plus some implausible excuse for not doing it herself...);
2. Barbara sends an email message to her friend Carol, responsible for the Book Shop front end, to reserve that book;
3. Carol receives Barbara email and sends a whatsApp message to Dave, the responsible of the Book Shop warehouse, to check the availability of the book and order it if necessary;
4. Dave checks if the book is available in the warehouse;
   (a) if it is,
      (i.) he sends a whatsApp message to Emily who is in charge for physically managing the books and informing the clients if their requests can be satisfied immediately;
      (ii.) Emily takes the book to the front end and sends a confirmation email to Barbara telling that the book is already there;
   (b) otherwise,
      i. Dave sends an email to Frank, the point of contact for the publisher of the required book, and orders it;
      ii. Frank sends a confirmation to Barbara via whatsApp telling her that the book will be available in two days.

For readability, we express the protocol using a more compact and formal syntax where $a \overset{mns,cnt}{\Longrightarrow} b$ stands for "agent a sends a message with content cnt via communication means mns to agent b". The symbol: stands for the prefix operator ($int{:}P$ is the protocol whose first allowed interaction is int, and the remainder is the protocol P) and has precedence over the other operators, $\vee$ stands for mutual exclusiveness, and ϵ represents the empty protocol:

$$\begin{aligned}P1 = {} & alice \overset{wa,buy}{\Longrightarrow} barbara{:}barbara \overset{em,reserve}{\Longrightarrow} carol{:}carol \overset{wa,checkAvail}{\Longrightarrow} dave{:}\\ & (dave \overset{wa,take2shop}{\Longrightarrow} emily{:}emily \overset{em,availNow}{\Longrightarrow} barbara{:}\epsilon \ \vee \\ & dave \overset{em,order}{\Longrightarrow} frank{:}frank \overset{wa,avail2Days}{\Longrightarrow} barbara{:}\epsilon)\end{aligned}$$

P1 receives the unanimous appreciation whatever the research community. In fact, two very intuitive properties are met: 1. apart from the first one, the message that each agent sends is a reaction to the message it just received, and there is an evident cause-effect link between two sequential messages; 2. in case some mutually exclusive choice must be made, the choice is up to only one agent involved in the protocol, and hence it is feasible. These good properties take different names depending on the research communities and on the authors. The first one is named, for example, "sequentiality" [20], "connectedness for sequence" [35], "explicit causality" [37]; the second "knowledge for choice" [20], "local choice" [34], "unique point of choice" [35]. Meeting these two properties is closely related to the absence of covert channels; they ensure that all communications between different participants are explicitly stated, and rule out those protocols whose implementation or enactment relies on the presence of secret/invisible communications between participants: a protocol description must contain all and only the interactions used to implement it [20].

The Bad. Those protocols that do not respect the connectedness for sequence and unique point of choice properties are bad, and it is not difficult to see why. Let us consider protocol $P2$:

$$P2 = alice \overset{wa,buy}{\Longrightarrow} barbara{:}carol \overset{wa,checkAvail}{\Longrightarrow} dave{:}\\ (dave \overset{wa,take2shop}{\Longrightarrow} emily{:}\epsilon \;\vee\; frank \overset{wa,avail2Days}{\Longrightarrow} barbara{:}\epsilon)$$

The protocol states that *carol* can send a *checkAvail* message to *dave* only after *alice* has sent a *buy* message to *barbara*, but how can *carol* know if and when *alice* sent that message? Also, the protocol states that either *dave* sends a message to *emily*, or *frank* sends a message to *barbara*: how can *frank* know if he is allowed to send a message to *barbara*, without coordinating with *dave* via some covert channel not shown in the protocol?

The Ugly. Protocols which are not syntactically correct are ugly, and are ignored by all the authors. However, some protocols may be ugly even if they are syntactically correct, for example if they are characterized by:

- "Causality unsafety": consider the two shuffled sequences $carol \overset{wa,buy}{\Longrightarrow} dave{:}\epsilon$ $|alice \overset{wa,buy}{\Longrightarrow} barbara{:}\epsilon$, where $|$ models interleaving (a.k.a. shuffling) between two protocol branches; suppose we are only able to observe what *alice* sends, and what *dave* receives. If *alice* sends *buy* and *dave* receives *buy*, we might think that the protocol above is respected. However, that observation might be due to *alice* sending *buy* to *dave*, which is not an allowed interaction: the protocol is not causality safe [35].
- "Non-Determinism": given an interaction taking place in some protocol state, we might want to deterministically know how to move to the next state. For example, if *alice* asks her mother to buy a book, and the protocol is

$$alice \overset{wa,buy}{\Longrightarrow} barbara : barbara \overset{wa,reserve}{\Longrightarrow} carol : \epsilon \;\vee\\ alice \overset{wa,buy}{\Longrightarrow} barbara : barbara \overset{wa,buyItYourself}{\Longrightarrow} alice : \epsilon$$

  we could move on either branch. If we opt to move on the first branch, the next expected action is that *barbara* asks *carol* to reserve a book. If, instead, *barbara* tells *alice* to buy the book by herself, we have to backtrack to the previous protocol state in order to check that this interaction is allowed as well; this is extremely inefficient and should be avoided [7]. While the notions of good and bad protocols are universally recognized, ugliness also depends on the formalism and its expressive power.

Can We Simply Ignore Bad Protocols? Let us suppose that the protocol is used for monitoring purposes: it does not need to be implemented or enacted. The agents in the MAS are already there, and they are heterogeneous black boxes behaving according to their own policies and goals, in full autonomy. The monitor observes messages that agents exchange, in a completely non obtrusive way, checks their compliance with the protocol ruling the MAS, and reports violations to some other entity or to the human(s) in charge for the security and safety of the monitored system.

Let us also suppose that the MAS protocol is $P1$ and the human(s) who set up the monitoring process know in advance that email messages cannot be observed by the monitor, for infrastructural, legal, or other reasons. Keeping interactions taking place via email in the protocol that the monitor will check would lead to false positives, as the monitor would look for messages foreseen by the protocol that it cannot see and would hence detect a protocol violation, but removing them from $P1$ leads to $P2$: a bad protocol! If the monitor observation ability is not perfect – which is an extremely realistic situation – there is no gain in struggling against bad protocols: unobservable interactions are there and generate the same problems of covert channels. We may state that what is "bad" in this situation is the monitor, since it is not able to observe everything should be observed, and not the AIP. This is another way to analyze the situation, but the problem still remains: a monitor with imperfect observation capability cannot be driven by a "perfect" protocol which includes unobservable messages. Rather, it might need to be driven by what we classified a "bad" protocol, describing all and only the messages that the monitor can actually observe.

Since we cannot ignore bad protocols, we opted for deepening our acquaintance with them. We developed RIVERtools, a tool able to (1) generate bad protocols starting from good ones with known unobservable channels, to correctly drive monitors with limited observation ability, and (2) explore the usage of bad protocols for Distributed Runtime Verification under suitable conditions. The next two sections introduce trace expressions and the algorithm implemented by RIVERtools to address the first point above. The second issue, namely how to partially decentralize the runtime monitoring process also when the AIP is bad, has been addressed in [28].

3 Background: Modeling AIPs with Trace Expressions

The general mechanism we propose for taking unobservable events into account during MAS monitoring, discussed in Sect. 4, can be applied to any formalism. However, to make our proposal more practical, we will discuss the algorithm we implemented for a specific formalism, trace expressions [3,5,7], and we briefly introduce it in the sequel. Trace expressions are based on the notions of *event* and *event type*. We denote by $\mathcal{E}$ the fixed universe of events subject to monitoring. An event trace $\bar{e}$ over $\mathcal{E}$ is a possibly infinite sequence of events in $\mathcal{E}$, and a trace expression over $\mathcal{E}$ denotes a set of event traces over $\mathcal{E}$. Trace expressions are built on top of event types (chosen from a set $\mathcal{ET}$), each specifying a subset of $\mathcal{E}$.

The semantics of event types is specified by the function *match*: if e is an event, and ϑ is an event type, then $match(e, \vartheta)$ holds if and only if event e matches event type ϑ; hence, the semantics $[\![\vartheta]\!]$ of an event type ϑ is the set $\{e \in \mathcal{E} \mid match(e, \vartheta) \text{ holds}\}$.

A trace expression $\tau \in \mathcal{T}$ represents a set of possibly infinite event traces, and is defined on top of the following operators:

- ϵ (empty trace), denoting the singleton set $\{\epsilon\}$ containing the empty event trace ϵ.
- $\vartheta{:}\tau$ (*prefix*), denoting the set of all traces whose first event e matches the event type ϑ, and the remaining part is a trace of τ.
- $\tau_1{\cdot}\tau_2$ (*concatenation*), denoting the set of all traces obtained by concatenating the traces of τ_1 with those of τ_2.
- $\tau_1{\wedge}\tau_2$ (*intersection*), denoting the intersection of the traces of τ_1 and τ_2.
- $\tau_1{\vee}\tau_2$ (*union*), denoting the union of the traces of τ_1 and τ_2.
- $\tau_1|\tau_2$ (*shuffle*), denoting the set obtained by shuffling the traces of τ_1 with the traces of τ_2.

Trace expressions support recursion through cyclic terms expressed by finite sets of recursive syntactic equations, as supported by modern Prolog systems. If $match(alice \overset{wa,buy}{\Longrightarrow} barbara, buy)$, $P4 = buy{:}(\epsilon \vee P4)$ denotes the protocol where *alice* may send one *buy* request to *barbara* and either terminate (ϵ) or start the protocol again ($\vee\ P4$). The traces denoted by $P4$ are
$\{alice \overset{wa,buy}{\Longrightarrow} barbara,\ alice \overset{wa,buy}{\Longrightarrow} barbara\ \ alice \overset{wa,buy}{\Longrightarrow} barbara,\ ...,\ (alice \overset{wa,buy}{\Longrightarrow} barbara)^n,\ ..., (alice \overset{wa,buy}{\Longrightarrow} barbara)^\omega\}$
namely, traces consisting of n instances of event $alice \overset{wa,buy}{\Longrightarrow} barbara$, with $n \geq 1$, plus the infinite trace. Infinite traces allow us to model systems that ought not to terminate, such as those involving a "service provider agent" that must be always be ready to answer the requests of its clients. We represent the infinite trace in an explicit way, to distinguish it from finite traces of any length.

Some Considerations on Events and Event Types. For presentation purposes, the protocols shown in Sect. 2 did not include event types but events. A simplified variant of $P1$ with event types is $P5$, where

$$P5 = bookReservationReq : availabilityCheckReq :$$
$$(availableNow : \epsilon\ \vee\ order : okOrder : available2Days : \epsilon)$$

and, for example,

$match(barbara \overset{em,reserve}{\Longrightarrow} carol, bookReservationReq)$
$match(barbara \overset{wa,reserve}{\Longrightarrow} carol, bookReservationReq)$
$match(carol \overset{wa,checkAvail}{\Longrightarrow} dave, availabilityCheckReq)$
$match(hillary \overset{wa,checkAvail}{\Longrightarrow} dave, availabilityCheckReq)$

The sequence $barbara \overset{em,reserve}{\Longrightarrow} carol \quad hillary \overset{wa,checkAvail}{\Longrightarrow} dave$ (namely $barbara \overset{em,reserve}{\Longrightarrow} carol$ followed by $hillary \overset{wa,checkAvail}{\Longrightarrow} dave$) is correct w.r.t. $P5$ since the first event matches $bookReservationReq$, after which an event matching $availabilityCheckReq$ is expected and in fact the second event matches it. Clearly, this sequence of messages does not make sense: we would like to state that the receiver of the first message must be the sender of the second one, but we cannot express such a constraint if the event type and protocol languages do not support variables. Parametric trace expressions [8] overcome this limitation by introducing parameters in the trace expression formalism. For space constraints we do not discuss them, but parameters are the element of the formalism that allows us, for example, to support multiple concurrent conversations and correctly track which message instance belongs to which conversation. The algorithm presented in Sect. 4 works for parametric trace expressions as well.

As far as observability is concerned, an event type $bookReservationReq$ that models reservation requests whatever the communication means used to send them (whatsApp or email) makes sense in most situations, apart those where the observability of messages is different depending on the communication means.

The sequence $barbara \overset{em,reserve}{\Longrightarrow} carol \quad carol \overset{wa,checkAvail}{\Longrightarrow} dave$ is correct w.r.t. $P5$, and the same holds for $barbara \overset{wa,reserve}{\Longrightarrow} carol \quad carol \overset{wa,checkAvail}{\Longrightarrow} dave$, but they might lead to different monitoring outcomes if the likelihood of $barbara \overset{em,reserve}{\Longrightarrow} carol$ to be observed by the monitor is different from that of $barbara \overset{wa,reserve}{\Longrightarrow} carol$.

In Sect. 4 we will limit our investigation to non-deterministic and contractive trace expressions (for example, definitions like $P = P \vee P$ are not contractive, as there is no means to "consume" interactions from P while rewriting it) and we will assume that event types model only sets of events whose observability likelihood is equivalent.

4 Partial Observability: When the Good Becomes Bad

In this section we discuss how a good protocol may become a (possibly) bad one, due to unobservability or partial observability of events. The human beings in charge for the system monitoring process may use the algorithm we present if they are aware of limited observability of events or channels, in order to generate a monitor which will check the system's compliance with the protocol output by the algorithm (possibly different from the original one, and possibly "bad") and avoid raising false alarms. Given a trace expression modeling an AIP, the monitor that dynamically verifies it can be generated automatically both for Jade [14] and for Jason [16], as discussed in [17] and [5] respectively.

For sake of readability, let us suppose that we are in charge for the monitoring process. We must associate with each event foreseen by the protocol, its "observability likelihood", namely the likelihood that the event can be observed by the monitor. Of course, we must know this parameter, which depends on the system to be monitored and on its context. If, when the event takes place, the monitor can always observe it, we associate 1 with the event. If the monitor can never observe the event (for example, the monitor can sniff whatsApp messages only, and the event is an email message), we associate 0 with it. If the event is transmitted over an unreliable or leaky channel, we may associate a number between 0 and 1, excluding the extremes, with it.

Let us consider $P1$ again, and let us suppose that:

(1) the observability likelihood of messages exchanged via email is 0;
(2) the observability likelihood of whatsApp messages sent by *frank* is 0.95;
(3) the observability likelihood of the other whatsApp messages is 1.

Condition 1 forces us to remove all messages exchanged via email from the protocol, and condition 3 forces us to keep all the other whatsApp messages but those sent from *frank*. The first and last conditions would lead to protocol $P2$. The second condition, however, requires a special treatment. In fact, message $\mathit{frank} \overset{wa,okOrder}{\Longrightarrow} \mathit{dave}$ could be either observed or not and both cases would be correct, even if the first one should be much more frequent than the second.

The subprotocol where $\mathit{frank} \overset{wa,okOrder}{\Longrightarrow} \mathit{dave}$ can either take place or not can be modeled by $\mathit{frank} \overset{wa,okOrder}{\Longrightarrow} \mathit{dave} : \epsilon \ \vee \ \epsilon$. The transformation from $P1$ to the protocol which takes observability likelihood into account requires the following steps:

(1) since the observability likelihood of messages exchanged via email is 0, remove them by $P1$;
(2) since the observability likelihood of the whatsApp message sent by *frank* is 0.95, substitute it with the corresponding subprotocol where the message can take place or not, and concatenate this subprotocol with the remainder;
(3) since the observability likelihood of the other whatsApp messages is 1, keep them all.

The result is

$$\begin{aligned} P3 = {} & \mathit{alice} \overset{wa,buy}{\Longrightarrow} \mathit{barbara} : \mathit{carol} \overset{wa,checkAvail}{\Longrightarrow} \mathit{dave} : \\ & (\mathit{dave} \overset{wa,take2shop}{\Longrightarrow} \mathit{emily} : \epsilon \ \vee \ (\mathit{frank} \overset{wa,okOrder}{\Longrightarrow} \mathit{dave} : \epsilon \ \vee \ \epsilon) \ \cdot \ \epsilon) \end{aligned}$$

which can be simplified into the equivalent protocol

$$\begin{aligned} P3' = {} & \mathit{alice} \overset{wa,buy}{\Longrightarrow} \mathit{barbara} : \mathit{carol} \overset{wa,checkAvail}{\Longrightarrow} \mathit{dave} : \\ & (\mathit{dave} \overset{wa,take2shop}{\Longrightarrow} \mathit{emily} : \epsilon \ \vee \ (\mathit{frank} \overset{wa,okOrder}{\Longrightarrow} \mathit{dave} : \epsilon \ \vee \ \epsilon)) \end{aligned}$$

Since dealing with likelihoods in $(0, 1)$ results into a more complex protocol, as the original protocol must be extended with the choice between observing the event or not, we might want to collapse likelihoods greater than a given threshold to 1, to avoid proliferation of choices: we may set a threshold above which events will be considered fully observable. Let Th be such threshold and P be the protocol to transform.

P' is obtained by P applying the following rules; L is the observability likelihood of interaction *int*

(1) if $L > Th$, *int* is kept;
(2) if $0 < L \leq Th$, *int* is transformed into the subprotocol where *int* can either take place or not, and suitably concatenated with the remainder;
(3) if $L = 0$, *int* is discarded.

Since different monitors might observe different events and observability might change over time, causing an evolution of the observable protocol, modeling the good global protocol and then transforming it based on contingencies is a better engineering approach than directly modeling the partial, observable protocol. However, even if we start from a good P protocol, P' might be bad or even ugly.

Observability-Driven Transformation of Trace Expressions. We implemented the algorithm sketched above for protocols modeled as trace expressions. The code has been developed in SWI-Prolog (http://www.swi-prolog.org/) and, for space constraints, is made available in a longer version of this paper available at http://www.disi.unige.it/person/MascardiV/Download/PAAMS-long18.pdf. The code for `filter_events` implementing the observability-driven transformation is 36 lines long and – provided a basic knowledge of logic programming – is self-explaining. Despite its simplicity, it can operate on very complex parametric and recursive (also non terminating) protocols. The magic behind the "invisible" management of non terminating protocols like $P4$ is the use of the SWI-Prolog coinduction library (http://www.swi-prolog.org/pldoc/doc/_SWI_/library/coinduction.pl) which allows to cope with infinite terms without entering into loops. RIVERtools [6,27] supports the development of AIPs modeled as trace expressions via a user-friendly GUI and implements static controls to check if a protocol is ugly, bad or good, suggesting how to decentralize the runtime monitoring process even in presence of bad protocols.

Experiments. We have experimented the filtering algorithm on a parametric trace expressions modeling the English Auction where the auctioneer proposes to sell an item for a given price and the bidders either accept or reject the proposal; as long as more than one bidder accepts, the price – which is a parameter of the protocol – is raised and another negotiation round is made. The protocol is consistent with the existing descriptions of the English Auction that can be found online, even if it slightly differs from the English Auction FIPA specification. The protocol description, as well as its code, can be downloaded from http://parametricTraceExpr.altervista.org/. We initially assumed that the only partially observable event was `buy(X)` with observability likelihood `0.5`. By setting the threshold to `0.7`, all occurrences of `buy(X)` became optional, while with a threshold equal to `0.4` they were all kept in the protocol. By setting the observability likelihood of `buy(X)` to `0`, any occurrence of `buy(X)` was removed from the protocol.

The filtering algorithm was also run on a variant of the Alternating Bit Protocol [25] with 6 agents. Different observability likelihoods and different thresholds were set with both protocols, to test the algorithm in an exhaustive way.

5 Related Work and Conclusions

To the best of our knowledge, MAS monitoring under partial or imperfect observability has been addressed in the context of normative multiagent organizations only, and by just a few works. Among them, [2] spun off from [18] and shows how to move from the heaven of ideal norms to the earthly condition of approximate norms. The paper focuses on conditional norms with deadlines and sanctions [39]; ideal norms are those that can be perfectly monitored given a monitor, and optimality of a norm approximation means that any other approximation would fail to detect at least as many violations of the ideal norm. Given a set of ideal norms, a set of observable properties, and some relationships between observable properties and norms, the paper presents algorithms to automatically synthesize optimal approximations of the ideal norms defined in terms of the observable properties. Even if the purpose of our work is in principle similar to that of [2,18], the approaches used to model AIPs are too different – also in expressive power – to compare them. A more recent work in the normative agents area is [24] that proposes information models and algorithms for monitoring norms under partial action observability by reconstructing unobserved actions from observed actions. While we assume to know in advance which events cannot be observed and we transform the ideal protocol into a monitorable one based on this information, the authors of [24] "guess" the actions that the monitor could not observe, but that must have taken place because of their visible effects. Outside the MAS community, partial ability of monitors to observe events is a well studied problem in many contexts including command and control [43] and runtime verification. In [38] the authors address the problem of gaps in the observed program executions. To deal with the effects of sampling on runtime verification, they consider event sequences as observation sequences of a Hidden Markov Model (HMM), and use an HMM of the monitored program to fill in sampling-induced gaps in observation sequences, and extend the classic forward algorithm for HMM state estimation to compute the probability that the property is satisfied by an execution of the program. Similarly to [24], that work complements ours by estimating the likelihood of an event to occur, whereas we assume to know that likelihood, and we transform the protocol – and hence the expected sequence of observed events – based on this knowledge. Other works pursuing the objective of suitably dealing with "lossy traces" in the runtime verification area are [13,33].

As part of our future work, we will evaluate the possibility to integrate our approach with those that complement it, like [24,38]. We expect that this might lead to some interesting result: on the observation (by the monitor) that the system is compliant with the bad AIP P', after many observations we might be able to state that "probably" the system is also compliant with the good AIP P. This research issue is very challenging: our roadmap involves deepening our understanding of the techniques that other researchers exploit to estimate the likelihood of unobserved events, and then merging those techniques into the algorithms supported by RIVERtools. We will also consider if and how our approach could be used to evaluate the protocol robustness to the presence of leaky and unreliable channels.

References

1. Alberti, M., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: The $\mathcal{S}$CIFF abductive proof-procedure. In: Bandini, S., Manzoni, S. (eds.) AI*IA 2005. LNCS (LNAI), vol. 3673, pp. 135–147. Springer, Heidelberg (2005). https://doi.org/10.1007/11558590_14
2. Alechina, N., Dastani, M., Logan, B.: Norm approximation for imperfect monitors. In: Proceedings of AAMAS 2014, pp. 117–124. IFAAMAS/ACM (2014)
3. Ancona, D., Bono, V., Bravetti, M., Campos, J., Castagna, G., et al.: Behavioral types in programming languages. Found. Trends Program. Lang. **3**(2–3), 95–230 (2016)
4. Ancona, D., Briola, D., Ferrando, A., Mascardi, V.: Global protocols as first class entities for self-adaptive agents. In: Proceedings of AAMAS, pp. 1019–1029. ACM (2015)
5. Ancona, D., Drossopoulou, S., Mascardi, V.: Automatic generation of self-monitoring MASs from multiparty global session types in Jason. In: Baldoni, M., Dennis, L., Mascardi, V., Vasconcelos, W. (eds.) DALT 2012. LNCS (LNAI), vol. 7784, pp. 76–95. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37890-4_5
6. Ancona, D., Ferrando, A., Franceschini, L., Mascardi, V.: Managing bad AIPs with RIVERtools. In: Demazeau, Y., et al. (eds.) PAAMS 2018, LNAI, vol. 10978, pp. 296–300. Springer, Cham (2018)
7. Ancona, D., Ferrando, A., Mascardi, V.: Comparing trace expressions and linear temporal logic for runtime verification. In: Ábrahám, E., Bonsangue, M., Johnsen, E.B. (eds.) Theory and Practice of Formal Methods. LNCS, vol. 9660, pp. 47–64. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30734-3_6
8. Ancona, D., Ferrando, A., Mascardi, V.: Parametric runtime verification of multiagent systems. In: Proceedings of AAMAS 2017, pp. 1457–1459. ACM (2017)
9. Ancona, D., Ferrando, A., Mascardi, V.: Improving flexibility and dependability of remote patient monitoring with agent-oriented approaches. In: IJAOSE. (2018, to appear)
10. Baldoni, M., Baroglio, C., Martelli, A., Patti, V.: Verification of protocol conformance and agent interoperability. In: Toni, F., Torroni, P. (eds.) CLIMA 2005. LNCS (LNAI), vol. 3900, pp. 265–283. Springer, Heidelberg (2006). https://doi.org/10.1007/11750734_15
11. Baldoni, M., Baroglio, C., Capuzzimati, F.: A commitment-based infrastructure for programming socio-technical systems. ACM Trans. Internet Techn. **14**(4), 1–23 (2014)

12. Baldoni, M., Baroglio, C., Capuzzimati, F., Micalizio, R.: Exploiting social commitments in programming agent interaction. In: Chen, Q., Torroni, P., Villata, S., Hsu, J., Omicini, A. (eds.) PRIMA 2015. LNCS (LNAI), vol. 9387, pp. 566–574. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25524-8_39
13. Basin, D., Klaedtke, F., Marinovic, S., Zălinescu, E.: Monitoring compliance policies over incomplete and disagreeing logs. In: Qadeer, S., Tasiran, S. (eds.) RV 2012. LNCS, vol. 7687, pp. 151–167. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-35632-2_17
14. Bellifemine, F.L., Caire, G., Greenwood, D.: Developing Multi-Agent Systems with JADE. Wiley, Hoboken (2007)
15. Bettini, L., Coppo, M., D'Antoni, L., De Luca, M., Dezani-Ciancaglini, M., Yoshida, N.: Global progress in dynamically interleaved multiparty sessions. In: van Breugel, F., Chechik, M. (eds.) CONCUR 2008. LNCS, vol. 5201, pp. 418–433. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85361-9_33
16. Bordini, R.H., Hübner, J.F., Wooldridge, M.: Programming Multi-Agent Systems in AgentSpeak Using Jason. Wiley, Hoboken (2007)
17. Briola, D., Mascardi, V., Ancona, D.: Distributed runtime verification of JADE multiagent systems. In: Camacho, D., Braubach, L., Venticinque, S., Badica, C. (eds.) Intelligent Distributed Computing VIII. SCI, vol. 570, pp. 81–91. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-10422-5_10
18. Bulling, N., Dastani, M., Knobbout, M.: Monitoring norm violations in multi-agent systems. In: Proceedings of AAMAS 2013, pp. 491–498. IFAAMAS (2013)
19. Casella, G., Mascardi, V.: West2East: exploiting web service technologies to engineer agent-based software. IJAOSE **1**(3/4), 396–434 (2007)
20. Castagna, G., Dezani-Ciancaglini, M., Padovani, L.: On global types and multiparty session. Log. Methods Comput. Sci. **8**(1) (2012)
21. Chopra, A.K., Christie, S., Singh, M.P.: Splee: a declarative information-based language for multiagent interaction protocols. In: Proceedings of AAMAS 2017, pp. 1054–1063. ACM (2017)
22. Chopra, A.K., Singh, M.P.: Cupid: commitments in relational algebra. In: Proceedings of AAAI 2015, pp. 2052–2059. AAAI Press (2015)
23. Cossentino, M.: From requirements to code with the PASSI methodology. Agent-Orient. Methodol. **3690**, 79–106 (2005)
24. Criado, N., Such, J.M.: Norm monitoring under partial action observability. IEEE Trans. Cybern. **47**(2), 270–282 (2017)
25. Deniélou, P.-M., Yoshida, N.: Multiparty session types meet communicating automata. In: Seidl, H. (ed.) ESOP 2012. LNCS, vol. 7211, pp. 194–213. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28869-2_10
26. Desai, N., Mallya, A.U., Chopra, A.K., Singh, M.P.: Interaction protocols as design abstractions for business processes. IEEE Trans. Softw. Eng. **31**(12), 1015–1027 (2005)
27. Ferrando, A.: RIVERtools: an IDE for runtIme VERification of MASs, and beyond. CEUR Workshop Proc. **2056**, 13–26 (2017)
28. Ferrando, A., Ancona, D., Mascardi, V.: Decentralizing MAS monitoring with DecAMon. In: Proceedings of AAMAS 2017, pp. 239–248. ACM (2017)
29. Francalanza, A., Pérez, J.A., Sánchez, C.: Runtime verification for decentralised and distributed systems. In: Bartocci, E., Falcone, Y. (eds.) Lectures on Runtime Verification. LNCS, vol. 10457, pp. 176–210. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75632-5_6

30. García-Ojeda, J.C., DeLoach, S.A., Robby: AgentTool III: from process definition to code generation. In: Proceedings of AAMAS 2009, pp. 1393–1394. IFAAMAS (2009)
31. Honda, K., Yoshida, N., Carbone, M.: Multiparty asynchronous session types. In: Proceedings of POPL 2008, pp. 273–284. ACM (2008)
32. Huget, M., Odell, J.: Representing agent interaction protocols with agent UML. In: Proceedings of AAMAS 2004, pp. 1244–1245. IEEE Computer Society (2004)
33. Joshi, Y., Tchamgoue, G.M., Fischmeister, S.: Runtime verification of LTL on lossy traces. In: Proceedings of SAC 2017, pp. 1379–1386. ACM (2017)
34. Ladkin, P.B., Leue, S.: Interpreting message flow graphs. Formal Aspects Comput. **7**(5), 473–509 (1995)
35. Lanese, I., Guidi, C., Montesi, F., Zavattaro, G.: Bridging the gap between interaction-and process-oriented choreographies. In: Proceedings of ICSEFM 2008, pp. 323–332. IEEE (2008)
36. Papazoglou, M.P.: Service -oriented computing: concepts, characteristics and directions. In: Proceedings of WISE 2003, p. 3. IEEE Computer Society (2003)
37. Singh, M.P.: Information-driven interaction-oriented programming: BSPL, the blindingly simple protocol language. In: Proceedings of AAMAS 2011, pp. 491–498. IFAAMAS (2011)
38. Stoller, S.D., Bartocci, E., Seyster, J., Grosu, R., Havelund, K., Smolka, S.A., Zadok, E.: Runtime verification with state estimation. In: Khurshid, S., Sen, K. (eds.) RV 2011. LNCS, vol. 7186, pp. 193–207. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29860-8_15
39. Tinnemeier, N.A.M., Dastani, M., Meyer, J.C., van der Torre, L.W.N.: Programming normative artifacts with declarative obligations and prohibitions. In: Proceedings of IAT 2009, pp. 145–152. IEEE Computer Society (2009)
40. Winikoff, M., Liu, W., Harland, J.: Enhancing commitment machines. In: Leite, J., Omicini, A., Torroni, P., Yolum, I. (eds.) DALT 2004. LNCS (LNAI), vol. 3476, pp. 198–220. Springer, Heidelberg (2005). https://doi.org/10.1007/11493402_12
41. Winikoff, M., Yadav, N., Padgham, L.: A new hierarchical agent protocol notation. Auton. Agent. Multi-Agent Syst. **32**(1), 59–133 (2018)
42. Yolum, P., Singh, M.P.: Commitment machines. In: Meyer, J.-J.C., Tambe, M. (eds.) ATAL 2001. LNCS (LNAI), vol. 2333, pp. 235–247. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45448-9_17
43. Yukish, M., Peluso, E., Phoha, S., Sircar, S., Licari, J., Ray, A., Mayk, I.: Limits of control in designing distributed C^2 experiments under imperfect communications. In: Military Communications Conference MILCOM 1994. IEEE (1994)

SimFI: A Transmission Agent-Based Model of Two Interacting Pathogens

Hélène Arduin(✉) and Lulla Opatowski

Biostatistics, Biomathematics, Pharmacoepidemiology and Infectious Diseases UMR 1181 Inserm - Université de Versailles Saint-Quentin-en-Yvelines - Institut Pasteur, Paris, France
helene.arduin@gmail.com, lulla.opatowski@pasteur.fr

Abstract. Mathematical modeling has become essential in the field of infectious diseases epidemiology, particularly when it comes to studying the transmission of pathogens. However, most models simulate only one pathogen, whereas many species can circulate in the same population and impact each other's transmission dynamics. Here we present a new agent-based model, SimFI, formalizing the co-circulation of two distinct pathogens in a human population. Several between-pathogen interaction mechanisms are implemented at the individual scale, and their effects on the global transmission dynamics at the population scale are studied. The model produces independent time series of infection cases for the two pathogens, mimicking the data usually collected by infectious diseases surveillance systems. This study highlights the importance of precisely representing phenomena occurring at the individual level and the complexity of ecological interactions, thus confirming the usefulness of agent-based models to better understand between-pathogen interactions in epidemiology.

Keywords: Agent-based model · Infectious diseases · Ecology Interactions · Epidemiology · Influenza

1 Introduction

Mathematical models have been widely used in epidemiology in recent years, particularly to describe the transmission dynamics of pathogens and diseases [1]. Classically, infectious diseases transmission models involve compartmental formalization based on ordinary differential equations and represent the transmission of only one pathogen, independently from all others [2]. However, pathogens transmission is a complex phenomenon which strongly depends on the involved population and pathogens. The simultaneous circulation of pathogens in a population might lead to non-neutral interactions affecting co-infected hosts, and leading to altered individual transmission dynamics [3,4].

A few compartmental models were proposed to study between-pathogen associations and better understand the mechanisms at stake [5,6]. Yet, these

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 72–83, 2018.
https://doi.org/10.1007/978-3-319-94580-4_6

approaches are limited as they impede the formalization of phenomena at the individual level, and do not allow for the integration of precise natural histories of diseases. Agent-based modeling is more appropriate to represent the complex processes occurring at the individual level, such as specific host-pathogen relationships and interactions between individuals or pathogens. In recent years, several agent-based models have been developed in the field of infectious diseases dynamics [7], and especially for influenza [8,9].

Here we present SimFI, a new stochastic agent-based model to simulate the co-circulation of two interacting pathogens in a human population. The model is specifically applied to the ecological interactions of the influenza virus.

2 Methods

The model description follows the ODD protocol as defined by Grimm *et al.* [10]. The model was programmed with the NetLogo software (version 5.2.1) [11], the code can be found here http://b2phi.inserm.fr/#/resources/71/NetLogo-SimFI-model. The R software (version 3.3.3) was used for all data analysis and graphics representations.

2.1 Ecological Context

The influenza virus infects the human respiratory tract, where it cohabits with many other viruses and bacteria. Three reviews report evidence of interactions between influenza and other pathogens, and propose hypothetical biological mechanisms [4,12,13].

These pathogens include many bacterial species, including natural colonizers of the human host such as *Streptococcus pneumoniae*, *Haemophilus influenzae* or *Staphylococcus aureus*. It has been suggested that the presence of the influenza virus can lead to a higher bacterial load and a longer carriage duration, due to increased bacterial cells' adherence and faster bacterial growth in the nasopharynx [14]. Macroscopically, this translates into altered acquisition and transmission dynamics of the bacteria in the population. There is also evidence of an increased risk for severe bacterial disease in the presence of influenza [15].

Influenza has also been suggested to interact with other influenza subtypes or other respiratory viruses, such as the rhinovirus or the respiratory syncytial virus [16,17]. On one hand, a primary infection with a virus may provide partial or total protection against other viruses, this mechanism is called cross-immunity [18]. On the other hand, infection with some viruses might boost influenza infection, increasing the severity of the disease [19].

2.2 Overview of the Model

Purpose. SimFI (Simulator of Flu in Interaction) aims at stochastically simulating influenza and a second pathogen in interaction in a virtual human population. Two independent versions were designed: one version simulating influenza in interaction with an epidemic-like pathogen (causing an annual outbreak and not circulating otherwise), and a second version with an endemic-like second pathogen (circulating in the population throughout the year, with no extinction).

Entities, State Variables and Scales

Agents and Their State Variables and Environmental Variables. There is one type of agents in this model: humans. A set of variables characterizes each individual of the population, it is presented in Table 1. A number of environmental variables are shared between all individuals, they are presented in Table 2 and detailed when necessary in the different sections.

Table 1. Individual variables.

Variable	Type	Update time
Age	Number of days	At each time step
Influenza-related		
Susceptible	Boolean	During state update
Infectious	Boolean	During state update
Immunized	Boolean	During state update
Incubation period time counter	Number of days	At each time step
Incubation period total duration	Number of days, drawn from $\Gamma\left(k = \frac{2^2}{0.1}; \theta = \frac{0.1}{2}\right)$, mean [20], arbitrary variance	At each new infection
Symptomatic period time counter	Number of days	At each time step
Symptomatic period total duration	Number of days, drawn from $\Gamma\left(k = \frac{4^2}{1}; \theta = \frac{1}{4}\right)$, mean [20], arbitrary variance	At each new infection
Second pathogen-related		
Susceptible	Boolean	During state update
Infected	Boolean	During state update
Infectious	Boolean	During state update
Immunized	Boolean	During state update
Asymptomatic period time counter	Number of days	At each time step
Asymptomatic period total duration	Number of days, drawn from $\Gamma\left(k = \frac{21^2}{25}; \theta = \frac{25}{21}\right)$ [21]	At each new contamination
Symptomatic period time counter	Number of days	At each time step
Symptomatic period total duration	Number of days, drawn from $\Gamma\left(k = \frac{12^2}{16}; \theta = \frac{16}{12}\right)$ [22]	At each new infection
Immunized period time counter	Number of days	At each time step

Scales. The model time step is the day. A simulation year lasts 364 days, in order to get 52 full weeks each year. The simulation year begins on October 1^{st}, the beginning of the winter season. The model is spatially explicit: each individual moves into the simulation space and is in contact with several other individuals each day, enabling pathogens' transmission. The simulation space is made of patches: two individuals are considered in contact if they are on the same patch at a given time step. An individual has 13 contacts per day on average, which is the average number of daily contacts per person reported in recent sociological studies carried out in different European countries [24]. Every day, the individuals move in the simulation space in a random direction at a distance randomly drawn between 1 and 10 patches.

Process Overview and Scheduling. During each time step, the following processes are executed in the following order.

Reset of the Incidence Counters for Both Pathogens. The number of new cases on each day and for each pathogen is called the incidence of the pathogen. The incidence counters are reset at each time step.

Influenza Outbreak Setup. On the first day of every year, the importation date of the first influenza cases is randomly drawn from a Gamma distribution, the influenza transmission rate is randomly drawn from a normal distribution, and a randomly chosen portion of the population is immunized against this year's influenza virus to account for acquired immunity. Then, on the day of the importation of the first influenza cases, a random number of individuals are infected by the virus and enter the incubation period. See Table 2 for default values of the different parameters.

Second Pathogen Setup. In the case of an epidemic second pathogen, the setup is similar to the one of influenza described above: the importation date of the first cases, the transmission rate, the number of immunized individuals, and the number of first cases can be randomly drawn in user-defined distributions. In the case of an endemic second pathogen, a portion of the population is randomly chosen to be colonized by the pathogen.

Pathogens Transmission. Depending on their current statuses regarding both pathogens, individuals can transmit and/or acquire one or both pathogens. The different statuses for a pathogen (also called its natural history) are detailed on Fig. 1. The pathogens' transmission probabilities vary according to the infectious statuses of the individuals in contact, and the interaction mechanisms that exist between the pathogens. The different interaction mechanisms and their implementation are detailed later on.

Aging and Renewal of the Population. Individuals get one day older at each time step, and die when they are 80 years old [23]. New individuals are randomly created in order to keep the total population size stable.

Table 2. Environmental parameters and their default values.

Description	Default value	Rationale
Population size	100,000 persons	Chosen for computational purposes
Life expectancy	80 years	[23]
Mean number of contacts	13 per individual per day	[24]
Maximum moving distance	10 patches	Empirically chosen
Acquisition interaction parameter	1	To be varied
Transmission interaction parameter	1	To be varied
Influenza-related		
Transmission rate	$\mathcal{N}(\mu = 0.033; \sigma = \frac{\mu}{30})$ per contact-day	Calibrated mean [25], fixed variance
Influenza importation date (each year)	$\Gamma\left(k = \frac{71^2}{28}; \theta = \frac{28}{71}\right)$	Idem
Reporting probability of a case	20%	Calibrated
Initial proportion of immunized individuals each year	23%	Calibrated
Shedding period	[incubation+1 day; symptoms+2 days]	[20]
Second pathogen-related		
Infection rate	4.2×10^{-5} per day	Calibrated [22]
Reporting probability of a symptomatic case	100%	Arbitrary
Shedding period	[day 0; symptoms+2 days]	[26]
Immunity after contamination	False	Arbitrary
Immunity duration	300 days	Arbitrary
Specific to an endemic pathogen		
Carriage rate	20%	[26]
Pathogenicity interaction parameter	1	To be varied
Specific to an epidemic pathogen		
Outbreak onset	$U(30; 60)$ days after influenza onset	To be varied according to the pathogen
Number of imported cases	$U(20; 30)$	Idem
Initial proportion of immunized individuals each year	25%	Idem
Cross-immunity interaction parameter	1	To be varied

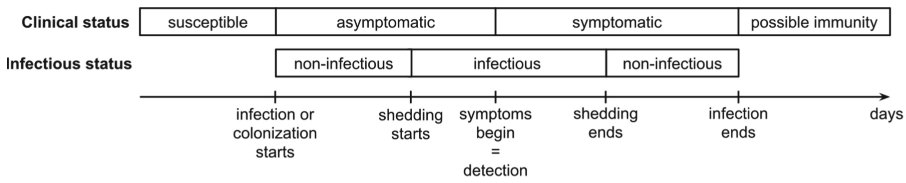


Fig. 1. Illustration of the generic natural history of a simulated pathogen in SimFI. When infected or colonized with a pathogen, an individual can go through an asymptomatic phase, followed by a symptomatic phase. During these two phases, the individual can be infectious or not according to the user-set parameters for the shedding duration. Upon recovery, the individual can be immunized for a user-set duration (the current simulated year for influenza for instance).

Synchronous Update of Infectious Statuses. The generic natural history of a simulated pathogen in SimFI is presented in Fig. 1. The duration of each phase of infection for both pathogens is monitored for every individual by a duration counter, incremented on each time step. When an individual reaches the maximum duration for a given infectious status, it is updated to the next logical infectious status (see Fig. 1). These updates and the ones from the transmission process all occur synchronously at the end of each time step.

2.3 Design Concepts

Emergence. Default values for the influenza-related parameters were chosen in order to represent the natural history of the virus and to produce incidence data similar to influenza-like illnesses data from the French surveillance network Sentinelles [25].

Interactions.

Interaction Mechanisms. Several interaction mechanisms are implemented in the model, macroscopically summarizing the main biological mechanisms described in the literature on influenza interactions presented earlier. They are implemented through multiplicative parameters modulating the transmission or infection probabilities of the pathogens (Fig. 2).

Acquisition Interaction: an individual's probability of acquiring a second pathogen when already infected by a first one is modulated by the α parameter. **Transmission Interaction:** the probability that a dually infected individual will transmit one pathogen to a susceptible person is modified by the θ parameter. **Cross-immunity Interaction:** only observed when two viruses are co-circulating (epidemic second pathogen scenario), the probability of acquiring one pathogen for individuals who are immunized against the other is changed by the μ parameter. It has been suggested that this mechanism has a time-limited effect (only during the first week of the immunized period for instance), another

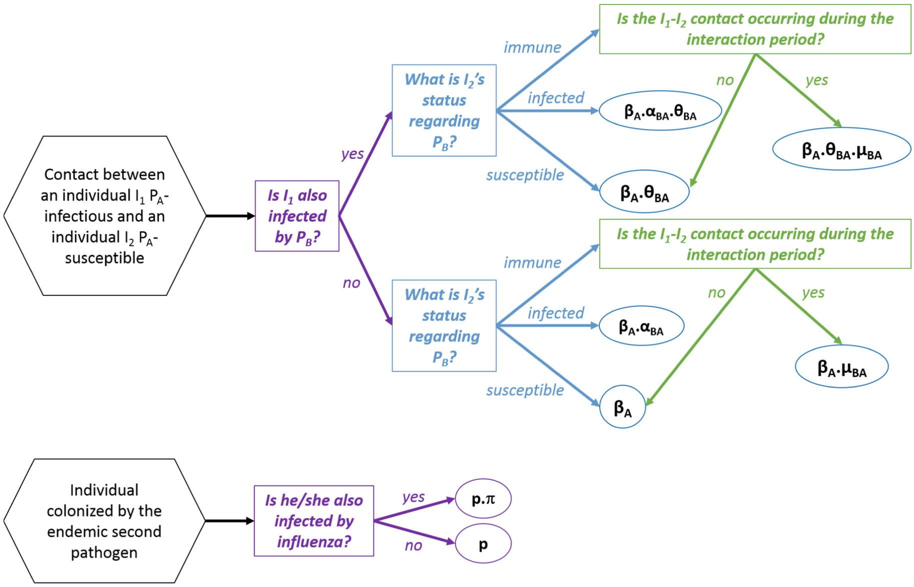


Fig. 2. Modulation of a pathogen's acquisition and infection probabilities due to the interaction mechanisms. (top) The transmission probability β_A of pathogen P_A is modulated by the infectious status of the two individuals in contact. α_{BA} is the parameter for the acquisition-interaction directed from P_B on P_A, θ_{BA} is the transmission-interaction parameter, and μ_{BA} represents the cross-immunity parameter. (bottom) The infection probability p of the endemic pathogen is modulated by the pathogenicity-interaction mechanism through the π parameter.

parameter thus allow to control this characteristic in the model. **Pathogenicity Interaction:** only proposed in the case of an epidemic and an endemic pathogens co-circulating, the probability to develop an infection with the endemic pathogen for individuals carrying both pathogens is altered by the π parameter.

Interaction Parameters. For all interaction mechanisms, interaction parameters larger than 1 lead to a higher risk of acquisition, transmission or infection in already-infected individuals. By opposition, parameters lower than 1 decrease this risk. A parameter equal to 1 corresponds to a neutral interaction, the risk remains unchanged.

The interaction mechanisms concerning acquisition, transmission and cross-immunity can represent an effect from influenza on the epidemiology of the second pathogen or the other way around. Therefore, there are two interaction parameters for each of these mechanisms, which can be independently activated or not. Regarding the pathogenicity interaction mechanism, only an effect from influenza on an endemic pathogen has been described, there is only one parameter for this mechanism.

Stochasticity. The model is stochastic, all infectious statuses durations for both pathogens are randomly drawn for each individual, transmission probabilities are also random. The Tables 1 and 2 detail the distribution laws that are used for each variable.

Observations. We record in SimFI the incidence (number of new cases) of influenza and of the second pathogen. For a given pathogen, we assume that a new case is reported when the infected individual starts to present symptoms.

2.4 Details

Initialization. First, the environmental variables are initialized as described in Table 2, then the individuals are created and their statuses initialized. Individuals are given random ages and spatial positions, and they are initially susceptible to both pathogens. If the second pathogen is endemic, a portion of the population is randomly chosen to be colonized.

3 Results

3.1 Simulated Data

An illustration of SimFI simulations is provided on Fig. 3. It shows the independent weekly counts of influenza new infections and the number of reported infections with the second pathogen, either endemic (left column) or epidemic (right column). In the baseline cases (Fig. 3A and F), the two pathogens circulate independently in the population. In the other cases (Fig. 3B-D and G-H), they are linked at the individual level through an interaction mechanism.

Influenza Simulation. The dynamics of influenza incidence are shown in red in Fig. 3. An outbreak emerges during each simulated year, unless the transmission probability (randomly chosen each year to reflect the variability of influenza strains causing outbreaks) is too low, in which case the virus goes extinct (rare occurrence, 0.14% of all simulated years). The mean duration of each outbreak (11.6 weeks) as well as the mean number of cases at the epidemic peak (609 cases per 100,000) are consistent with French incidence data from the Sentinelles surveillance network [25]. As the model is stochastic, differences between each outbreak exist, especially regarding their size, onset and duration. The mean annual number of influenza cases over the 3 000 simulated years is 5,260 per 100,000 ($[3{,}760–6{,}760]_{95\%\ \mathrm{CI}}$, 95% confidence interval), compared to the 4,240 annual cases on average in France.

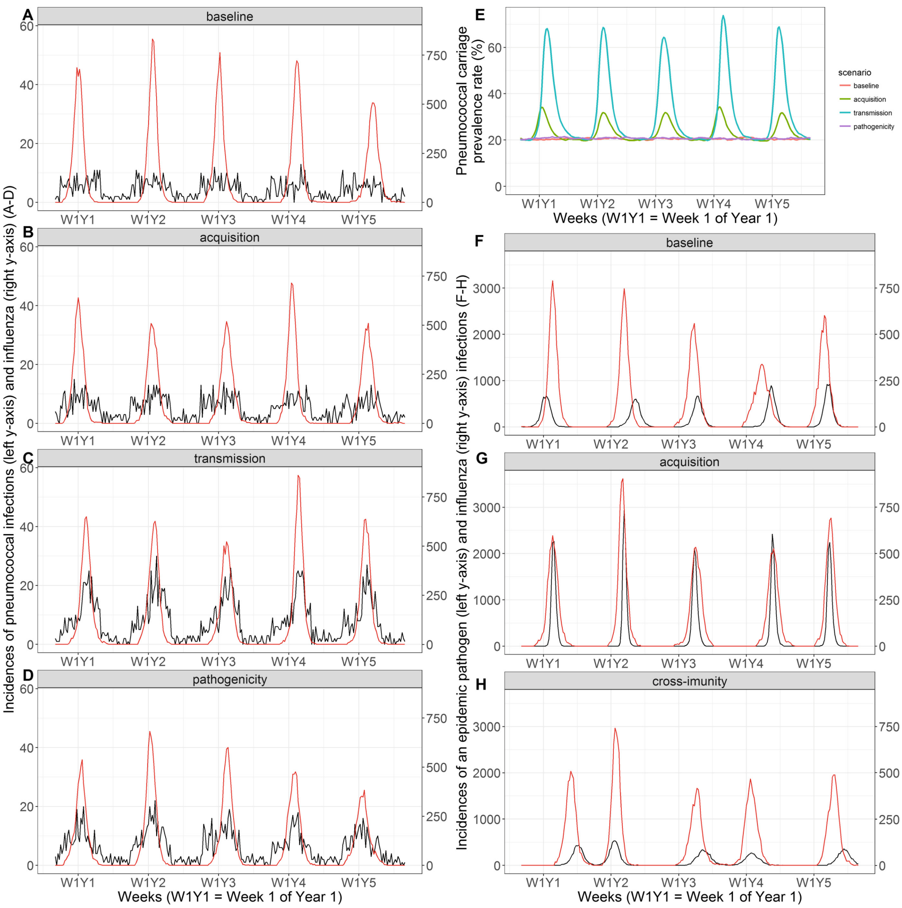


Fig. 3. Weekly incidence of simulated cases per 100,000 for the three possible pathogens. The influenza time series are presented in red (A–D; F–H). On the left-hand side, the black time series correspond to infection cases by an endemic second pathogen with no interaction mechanism activated (A), an acquisition-interaction of strength 50 (B), a transmission-interaction of strength 50 (C), and a pathogenicity-interaction of strength 50 (D). On the right-hand side, panel (E) represents the carriage in the population of the endemic second pathogen for the baseline scenario (orange), the acquisition-interaction of strength 50 (green), the transmission-interaction of strength 50 (blue), and the pathogenicity-interaction of strength 50 (purple). The black time series (F–H) correspond to an epidemic second pathogen with no interaction mechanism activated (F), an acquisition-interaction of strength 25 (G), and a cross-immunity-interaction of strength 0.8 (H). The represented data were chosen for five among the 3,000 simulated years for each scenario for their explicit representation of each interaction mechanism effect on infection dynamics. (Color figure online)

Simulation of an Endemic Second Pathogen. SimFI was calibrated to reproduce the incidence of pneumococcal infections in France (a few hundred cases per year [27]). Pneumococcal infections are caused by *Streptococcus pneumoniae* or pneumococcus, a commensal bacteria of the upper respiratory tract in humans. The simulations lead to an average of 242 per 100,000 $[241–243]_{95\%\ \mathrm{CI}}$ pneumococcal infections per year, with the vast majority of cases occurring during the winter period (203 cases per 100,000 on average between October and April). As observed on Fig. 3A–D, pneumococcal infections incidence series (in black) differ depending on the activated interaction mechanism. Notably, pneumococcal carriage in the population is strongly correlated to the influenza outbreaks in the case of the acquisition and transmission interaction mechanisms (Fig. 3E). This phenomenon is not reported in literature, thus scenarios leading to such data are not very realistic.

Simulation of an Epidemic Second Pathogen. Figure 3F–H represent incidence series of two successive epidemics during a simulated year. The first one (in red) is caused by the influenza virus as described before, the second (in black) is caused by an unspecified epidemic pathogen with arbitrary characteristics chosen to illustrate the model's behavior. Each interaction mechanism has a specific effect on the shape and timing of the second epidemic. If acquisition of the second pathogen is facilitated (Fig. 3G), a greater number of infections by this pathogen is recorded and the epidemic occurs faster than in the baseline scenario (Fig. 3F) If the cross-immunity between the two pathogens is amplified (Fig. 3H), the second epidemic is delayed by several weeks compared to the baseline scenario, and the annual number of infections is much lower.

4 Discussion

SimFI is an agent-based model simulating the co-circulation of two pathogens in interaction in a human population. This model can generate incidence data (cases counts), which is the type of data usually recorded by infectious diseases surveillance systems. Its main advantage is to relate interaction mechanisms at the individual level to the global dynamics of pathogens at the population level. The individual approach enables precise definition of the diseases natural histories in individuals, which is not possible with other modeling methods that necessitates to group individuals in subpopulations.

A wide range of values for the interactions parameters were assessed, from 2 to 100. This interval was defined based on published estimates from different studies and for different kind of infections [5,6]. In a previous study, we estimated from the analysis of French pneumococcal meningitis data [5] that respiratory viruses increased bacterial transmission by a 8.7 factor $[4.6–14.4]_{95\%\ \mathrm{CI}}$ in co-infected individuals, and increased the risk of developing a bacterial disease by a 92 factor $[28–361]_{95\%\ \mathrm{CI}}$. An American study by Shrestha *et al.* [6] analyzed two bacterial pneumonia datasets and found an increased susceptibility to pneumonia caused by the influenza infection by a 85 factor $[27\text{–}160]_{95\%\,\mathrm{CI}}$ or 115 factor $[70\text{–}230]_{95\%\,\mathrm{CI}}$ depending on the dataset.

Interestingly, in our simulations, the transmission and acquisition interaction mechanisms, which involve distinct biological mechanisms, were not differentiable from the resulting incidence time series. On the contrary, the interaction mechanism on disease pathogenicity lead to highly specific incidence curves.

As the focus of this study was between-pathogen interactions, and to keep the model simple, several SimFI features do not represent accurately real-world mechanisms. First, pathogens can infect humans differently depending on their age, due to variations of immunity and susceptibility. For instance, young children and the elderly are more at risk for severe influenza infections. In the future, integrating age-related characteristics of the diseases should improve the realism of the model. Second, the spatio-temporal contact network implemented induces completely random moves of individuals, and thus an unrealistic mixing. A more accurate representation could be implemented, eg. by incorporating age-specific movement patterns or favorite destinations (home, workplace, schools).

More generally, this model is easily adaptable to different pathogens, and more complex features could be added, such as an increased lethality induced by the disease, more elaborate mixing pattern between individuals, or other interaction mechanisms. We proposed here the simplest possible model including a variety of ready-to-use between-pathogen interactions mechanisms, providing a framework to simulate *in silico* realistic data of infection and test different interaction hypotheses.

References

1. Grassly, N.C., Fraser, C.: Mathematical models of infectious disease transmission. Nat. Rev. Microbiol. **6**(6), 477–487 (2008)
2. Qiu, Z., Feng, Z.: Transmission dynamics of an influenza model with vaccination and antiviral treatment. Bull. Math. Biol. **72**(1), 1–33 (2010)
3. Pawlowski, A., Jansson, M., Sköld, M., Rottenberg, M.E., Källenius, G.: Tuberculosis and HIV co-infection. PLoS Pathog. **8**(2), e1002464 (2012)
4. Bosch, A.A.T.M., Biesbroek, G., Trzcinski, K., Sanders, E.A.M., Bogaert, D.: Viral and bacterial interactions in the upper respiratory tract. PLoS Pathog. **9**(1), e1003057 (2013)
5. Opatowski, L., Varon, E., Dupont, C., Temime, L., van der Werf, S., et al.: Assessing pneumococcal meningitis association with viral respiratory infections and antibiotics: insights from statistical and mathematical models. Proc. Biol. Sci./Roy. Soc. **280**(1764), 20130519 (2013)
6. Shrestha, S., Foxman, B., Weinberger, D.M., Steiner, C., Viboud, C., et al.: Identifying the interaction between influenza and pneumococcal pneumonia using incidence data. Sci. Transl. Med. **5**(191), 191ra84 (2013)
7. Willem, L., Verelst, F., Bilcke, J., Hens, N., Beutels, P.: Lessons from a decade of individual-based models for infectious disease transmission: a systematic review (2006–2015). BMC Infect. Dis. **17**(1), 612 (2017)
8. Roche, B., Drake, J.M., Rohani, P.: An agent-based model to study the epidemiological and evolutionary dynamics of influenza viruses. BMC Bioinform. **12**(1), 87 (2011)

9. Guo, D., Li, K.C., Peters, T.R., Snively, B.M., Poehling, K.A., et al.: Multi-scale modeling for the transmission of influenza and the evaluation of interventions toward it. Sci. Rep. **5**, 8980 (2015)
10. Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., et al.: A standard protocol for describing individual-based and agent-based models. Ecol. Model. **198**(1–2), 115–126 (2006)
11. Wilensky, U.: NetLogo (1999). http://ccl.northwestern.edu/netlogo/
12. Mina, M.J., Klugman, K.P.: The role of influenza in the severity and transmission of respiratory bacterial disease. Lancet Respir. Med. **2**(9), 750–763 (2014)
13. Opatowski, L., Baguelin, M., Eggo, R.M.: Influenza interaction with cocirculating pathogens, and its impact on surveillance, pathogenesis and epidemic profile: a key role for mathematical modeling. PLoS Pathog. **14**(2), e1006770 (2017)
14. Short, K., Habets, M.: Interactions between Streptococcus pneumoniae and influenza virus: a mutually beneficial relationship? Future Microbiol. **7**(5), 609–624 (2012)
15. Lee, L.N., Dias, P., Han, D., Yoon, S., Shea, A., et al.: A mouse model of lethal synergism between influenza virus and Haemophilus influenzae. Am. J. Pathol. **176**(2), 800–811 (2010)
16. Goka, E.A., Vallely, P.J., Mutton, K.J., Klapper, P.E.: Single and multiple respiratory virus infections and severity of respiratory disease: a systematic review. Paediatr. Respir. Rev. **15**(4), 363–370 (2014)
17. Yoshida, L.M., Suzuki, M., Nguyen, H.A., Le, M.N., Vu, T.D., et al.: Respiratory syncytial virus: co-infection and paediatric lower respiratory tract infections. Eur. Respir. J. **42**(2), 461–469 (2013)
18. Walzl, G., Tafuro, S., Moss, P., Openshaw, P.J., Hussell, T.: Influenza virus lung infection protects from respiratory syncytial virus-induced immunopathology. J. Exp. Med. **192**(9), 1317–1326 (2000)
19. Schnepf, N., Resche-Rigon, M., Chaillon, A., Scemla, A., Gras, G., et al.: High burden of non-influenza viruses in influenza-like illness in the early weeks of H1N1v epidemic in France. PloS One **6**(8), e23514 (2011)
20. Punpanich, W., Chotpitayasunondh, T.: A review on the clinical spectrum and natural history of human influenza. Int. J. Infect. Dis. **16**(10), e714–e723 (2012)
21. Abdullahi, O., Karani, A., Tigoi, C.C., Mugo, D., Kungu, S., et al.: Rates of acquisition and clearance of pneumococcal serotypes in the nasopharynges of children in Kilifi District, Kenya. J. Infect. Dis. **206**(7), 1020–1029 (2012)
22. Chidiac, C.: Antibiothérapie par voie générale dans les infections respiratoires basses de l'adulte. Pneumonie aiguë communautaire. Exacerbations de bronchopneumopathie chronique obstructive. Médecine et Maladies Infectieuses **41**(5), 221–228 (2011)
23. INSEE: (2017). https://www.insee.fr/fr/statistiques/2383448
24. Mossong, J., Hens, N., Jit, M., Beutels, P., Auranen, K., et al.: Social contacts and mixing patterns relevant to the spread of infectious diseases. PLoS Med. **5**(3), 0381–0391 (2008)
25. Sentinelles: https://websenti.u707.jussieu.fr/sentiweb/?page=presentation
26. Abdullahi, O., Nyiro, J., Lewa, P., Slack, M., Scott, J.A.G.: The descriptive epidemiology of Streptococcus pneumoniae and Haemophilus influenzae nasopharyngeal carriage in children and adults in Kilifi district, Kenya. Pediatr. Infect. Dis. J. **27**(1), 59–64 (2008)
27. Haut Conseil de la Santé Publique: Avis relatif aux recommandations de la vaccination pour les adultes et les enfants âgés de plus de 2 ans à risque d'infection invasive à pneumocoque 1–13 (2013)

Electric Vehicles Fleet for Frequency Regulation Using a Multi-Agent System

Jean-Baptiste Blanc-Rouchossé[1(✉)], Anne Blavette[2], Guy Camilleri[1], and Marie-Pierre Gleizes[1]

[1] IRIT, Université de Toulouse, 118 Route de Narbonne, 31062 Toulouse, France
{Jean-Baptiste.Blanc-Rouchosse,camilleri,Gleizes}@irit.fr
[2] Univ. Rennes, SATIE, CNRS, Rennes, France
anne.blavette@ens-rennes.fr

Abstract. The production of PhotoVoltaic (PV) energy depends on the solar irradiance level. The PV power plant fluctuations may have a significant impact on the frequency regulation in sufficiently small power systems, such as islands. The objective of this paper is to present a method using cooperative multi-agent systems to reduce the frequency fluctuations due to the unpredicted fluctuations of the PV production using electric vehicles as electricity storage units in an isolated power system.

Keywords: Smart grids · PV power plants · Electric vehicles Multi-agent systems · Frequency regulation

1 Introduction

The portion increase of renewable energies in the energy mix may cause stability issues. Actually, the stability of the electric network depends on the balance between the generated power and the consumed power. In fact, the electric power is a flux: the incoming power in the network (generated power) and the outgoing power (consumed power) must be equal to insure the flux conservation. The kinetic energy stored in synchronous generators of producers connected to the network compensate the consumption/production discrepancy. This compensation entails a change in the kinetic energy, thus a variation of the generators speed and of the grid frequency. They increase if the production is higher than the consumption and decrease if the production is lower than the consumption. Hence, the stability is insured by speed regulation of these generators around its target speed. So the stability relies mainly, on a second-to-second basis, on the inertia of the synchronous generators connected to the network (primary frequency response). Therefore, the global power unbalance has to be minimised to reduce the kinetic energy variation, and the frequency deviation. Ideally, to do that, the consumption has to be forecast. Then, the production of power plants has to be planned a day-ahead in order to balance the forecasted consumption. However, some renewable power plants (PV and wind powered for

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 84–96, 2018.
https://doi.org/10.1007/978-3-319-94580-4_7

example) cannot be considered as other power plants since their production is poorly dispatchable. To control them, a part of the energy they could produce can be shedded. But this energy is then lost. So their production must be forecast: the manager of the PV power plant has to provide a day-ahead production commitment. The use of a storage equipment can be a solution to this issue. In fact, a virtual power plant can be defined by associating this storage equipment to a renewable power plant. In this way, when the production is lower than the forecast, the storage equipment can be discharged to counterbalance the discrepancy between production and consumption.

In this work, we propose a multi-agent system for regulating the frequency of an isolated electric grid including PV power plant and electric vehicles. The frequency is supposed to be unstable due to the presence of an unexpected gap between the actual production of a PhotoVoltaic (PV) power plant and the forecast production of this power plant. The frequency is regulated by defining the action of each vehicle taking into account criteria on the vehicle. The multi-agent architecture developed is mainly composed of vehicle agents corresponding to physical vehicles which are the considered storage means; and charging station agents which are associated to physical charging stations. Each EV agent does not communicate with other EV agents, but only with the charging station agents. This choice in addition to the cooperative attitude of agents enable to reduce the complexity of the solution, and therefore its efficiency.

This paper is organised as follows: firstly, some approaches are presented. Secondly, the models used in the simulation are introduced followed by the proposed multi-agent system architecture. Finally, some results obtained with this system are analysed.

2 Related Works

When there is an overproduction compared to the commitment, the storage equipment is recharged. In this way, this virtual plant should be able to respect its commitment [1–3]. However, this solution may require an important storage equipment capacity. Using EV batteries could be a solution to avoid to invest in batteries reserved to this application. Lund and Kempton [4] showed the interest of coupling EV batteries with high integration of wind power. These studies have been realised considering multiple management strategies of the fleet. However, the low energy capacity of EV batteries imposes to have an important number of vehicles. Each vehicle behaviour needs to be defined in order to solve the stability issue, so the frequency regulation. This problem is then complex due to the large amount of entities (variables) which are present.

To manage this complex issue, artificial intelligence approaches have been proposed (see [5]), including fully-distributed approaches based on multi-agent systems. These approaches are characterised by a high-level of flexibility, scalability and robustness while requiring a reduced computing time which makes them well-suited for solving large problems. Two major aspects should be considered to improve the performance of a multi-agent system: its architecture and the actions that the agents can undertake. Concerning the architecture, it has been shown in [6] that for a given problem, the architecture type achieving the best performances could vary as a function of the parameters of the problem, such as the electrical network conditions for instance.

However, to the best of the authors' knowledge, the multi-agent approach for the frequency regulation seems to be absent from the literature. Yet, this approach has been considered widely in the context of smart grids. Many of these studies consider the power regulation issue [7,8]. Paniah et al. [7] propose a solution consisting in associating an agent to each actor: EV, power plants, etc. and some additional agents are used to realise a particular function: combine power plants, produce production schedules, etc. The different agents communicate by exchanging schedules and negotiate between each other until they find a solution respecting all the constraints. These provisional schedules are adjusted during operation to adapt the system to the production hazards. Hernandez et al. [8] include in the multi-agent system a forecast of energy demand. This forecast is necessary to manage the energy in the virtual power plant considering the upcoming needs.

More generally, several studies have already been carried out on the management of a vehicle fleet to respond to smart grid issues: respect of a power profile [1–3], frequency stability [9,10], etc. Different methods have been used to manage this fleet in spite of the complexity. In the case of the frequency regulation [9], a first solution is to treat the issue locally. A term proportional to the frequency deviation is added to the charging reference power of the vehicle [9]. When there is an overproduction, therefore a frequency increase, the vehicle increases its consumption and decreases it when there is an underproduction. However, this method does not take into account criteria on the vehicle such as constraint on its state of energy (ratio between the stored energy and the capacity). Another possibility is to recharge vehicles by default. Then, in the case of a fall of production, vehicles are disconnected from the grid or are discharged to the grid if their state of energy is greater than a predefined threshold [10]. Other methods use an aggregator to solve a global issue such as the minimisation of the cost due to different factors such as operating cost of generators and vehicle-to-grid services [11]. These two solutions take into account the criterion on the state of energy of vehicles. But the ageing issue of the batteries has not been discussed.

3 Model of the Grid

For this study, the gap between production and consumption is supposed to come from the PV power plant and from the consumption of the fleet. In fact, synchronous generators speed is not supposed to be regulated to counteract second-to-second fluctuations generated by the PV power plant. This forces the EV fleet to be the sole responsible for frequency regulation on a second-to-second basis. However, it is considered that the EV fleet energy needs are provided by both the synchronous generator and the PV power plant on a day-to-day basis. In this way, the gap between produced power (P_{prod}) and consumed power (P_{cons}) is fully offset by the kinetic energy variation of the synchronous machines: $\Delta P = P_{prod} - P_{cons} = \eta \frac{dE_k}{dt}$, with η the energy efficiency of generators, $E_k = Kf^2$ the kinetic energy of all the synchronous generators, K (set to 3.10^5 kW.s^3) the image of their inertia and f the grid frequency.

In this study, the power gap is composed of two terms: $\Delta P = \Delta P_{PV} - P_{fleet}$, with ΔP_{PV} the production gap of the PV power plant. In fact, the power of the radiant flux perceived by the PV panels, the irradiance, is supposed to be perfectly forecast at a 30 s time step, which is the minimum limit reached by the best available techniques. The actual average irradiance over 3 s is therefore equal to the forecast. However, the instant irradiance (Irr) is not equal to this forecast ($Irr_{forecast}$): it leads to a production gap ΔP_{PV} supposed proportional to the irradiance gap ($\Delta P_{PV} = \eta_{PV} \cdot S \cdot (Irr_{forecast} - Irr)$, $\eta_{PV} = 0.1$ the energy efficiency of the PV power plant, $S = 10^6 \text{m}^2$ its surface). This is a simple model but it should not influence the presented method. As regards P_{fleet}, it is the total power consumed by the fleet of vehicles connected to the grid. EVs are supposed to be connected when they are not travelling. All the vehicles make two trips a day: the journey to work between 8 am and 10 am and the return trip between 5 pm and 7 pm [12]. In this paper, the trips distance are considered equal to 35 km ± 10%. The departure time of each vehicle is supposed to be known.

The first constraint of the problem is to maintain the frequency in the allowed interval [49.5 Hz; 50.5 Hz]. To do so, each EV state needs to be defined: recharging, discharging or idling. However, constraints on the EV need to be taken into account. Firstly, the vehicle has to be sufficiently charged at departure time. In other words, the battery State of Energy (SoE), i.e. the ratio between the energy stored in the battery ($E_{bat}(t)$) and its capacity (E_{bat}^{cap}), $SoE(t) = \frac{E_{bat}(t)}{E_{bat}^{cap}}$, has to be greater than a threshold SoE_{min}: $SoE(t_{departure}) \geq SoE_{min}$. The state of energy has been simulated by the equation $SoE(t + \Delta t) = SoE(t) + \Delta SoE(t)$, with $\Delta SoE(t)$ defined in Eq. 1, and with α the recharge and discharge power loss rate. The power $P_{ev}(t)$ is the power absorbed from the network, with $P_{ev}(t) > 0$ if the vehicle is recharging and $P_{ev}(t) < 0$ if it is discharging.

$$\Delta SoE(t) = \frac{\Delta t \cdot (1 - sign(P_{ev}(t)) \cdot \alpha) \cdot P_{ev}(t)}{E_{bat}^{cap}} \quad (1)$$

A second objective on the vehicles is to limit the ageing of the battery due to these additional solicitations. Most of lithium-ion batteries ageing models are based on the concept of half-cycle i.e. the phase during which the battery performs a recharge or a discharge. The Depth of Discharge (DoD) of the half-cycle is the difference between the final SoE and the initial SoE [1]. For exemple, if a battery has its SoE at 0.6 (charged at 60%), and is discharged until its SoE gets to 0.4 (charged at 40%), then the DoD of this half-cycle is −0.2. To identify the cycles, the instantaneous depth of discharge ($DoD(t)$) has been used. It corresponds to the depth of discharge of the current half-cycle if it ended at that instant (Eq. 2).

$$DoD(t+\Delta t) = DoD(t) + \Delta SoE(t) \text{ if } DoD(t) \cdot P_{ev}(t) \geq 0, \text{ else } DoD(t+\Delta t) = 0 \quad (2)$$

In the study, decisions are taken at a very short time (one second). Hence, micro-cycles of recharge and discharge of few seconds can appear. To avoid to excessively speed up the batteries ageing, an objective on the depths of discharge has been taken into account: it consists in maintaining the battery in half-cycles of recharge or discharge of about 30 s minimum.

4 Implemented System Based on Multi-Agent Systems

In order to achieve the best performances for regulating dynamically energy networks, we adopted an Adaptive Multi-Agent System approach (AMAS). This approach aims to propose solutions applicable to complex systems. It is based on the concept of emergence, concept that makes it possible to explain phenomena difficult to decompose: these phenomena are both a consequence of the parts that make up the system and of their interactions [13]. Thus, the aim of an AMAS is to have the required functionality which emerges from the behaviour of agents: autonomous and active entities in an environment, having a goal and cooperating with each other. To design the AMAS, the ADELFE toolkit has been used [14].

4.1 Environment and Agents Identification

The environment of the multi-agent system in this application is the electric grid, the producers and the consumers: the PV power plant and the fleet of EV respectively.

Each agent corresponds to a domain entity. Among the possible entities, there are vehicles, charging stations, buses, power lines and producers. The objective is to have, just as in [15], an agent to control each constraint: about the frequency, the batteries state of energy and the cycles depth of discharge. For the batteries state of energy constraint, it is natural to match an agent with each vehicle. For the frequency constraint, an agent has been associated to the charging stations. Indeed, the charging stations has access to the frequency. Moreover, vehicles are connected to the charging stations: they should be able to communicate in order to cooperate. Moreover, the number of interactions between agents is limited, because a vehicle agent cannot communicate with the other vehicle agents.

Charging Stations. The agents associated with a charging station aim at keeping the frequency of the network in the allowed interval by sending cooperation requests to connected vehicles: increase or decrease of the consumption.

Electric Vehicles. The objective of this agent is to ensure that the constraints regarding its vehicle are satisfied. These agents act by defining the action to applied to the battery: recharge, discharge or idle, according to the vehicle state, the departure time and the cooperation request received from the charging station agent.

4.2 Agents Behaviour

In an AMAS, agents are autonomous and have cooperative attitude: an agent acts in order to help another agent if the latter encounters difficulties in achieving its goal [16]. To this end, it needs to have quantitative criteria of comparison: the criticality. A normalised criticality between 0 and 1 is associated to each objective or constraint of the agents. A 1 criticality value means that the agent is far away from its objective. If it reaches its goal, criticality is equal to zero.

Criticality on the Minimum State of Energy. This criticality, defined by Eq. 3, depends on SoE_{min}, the minimum state of energy required by the driver and t_r, the remaining time before departure. t_{min} is the minimum time required to reach the requested SoE. Thus, if $t_{min} = t_r$, the vehicle has to be recharged until departure: the criticality is maximum. In return, if $t_{min} << t_r$, the vehicle can postpone its recharge if necessary. It can afford to cooperate, even by discharging itself: the criticality has to be low.

$$Cr_{SoE} = \frac{t_{min}}{t_r} \text{ if } 0 \leq t_{min} \leq t_r, \text{ with } t_{min} = \frac{(SoE_{min} - SoE(t)) \cdot E_{bat}^{i,cap}}{P_{ev,max}} \tag{3}$$

Criticality on the Depth of Discharge. In Eq. 4, DoD_{ref} is the depth of discharge for which the criticality is null. This value has been set to have a null criticality for a 30-s cycle for a 30 kWh battery with a nominal 20 kW power rating. The choice of 30 s has been made in such a way that the battery does not suffer from cycles more dynamic (i.e. frequent and deep) than when the vehicle is used for transport purposes only, with respect to the FTP75[1] profile.

$$Cr_{DoD} = \left(1 - \frac{|DoD(t)|}{DoD_{ref}}\right) \text{ if } DoD(t) \neq 0,\ Cr_{DoD} = 0 \text{ otherwise} \tag{4}$$

Criticality on Frequency. In Eq. 5, f is the grid frequency, f_0 the nominal frequency (50 Hz) and Δf the allowed frequency deviation (0.5 Hz). This criticality is low when the frequency is close to 50 Hz and increases until reaching its maximum at the limit of the allowed interval.

$$Cr_f = \frac{|f - f_0|}{\Delta f} \text{ if } |f - f_0| \leq \Delta f \tag{5}$$

Charging Station Behaviour. The charging station agent receives the grid frequency. It deduces the value of the criticality defined previously (Cr_f). From this frequency, the agent is able to know if there has been an unbalance between production and consumption. Then, it sends a cooperation request to connected vehicles. It also defines two criticality thresholds ($Th(discharge)$ and $Th(postpone)$) that helps the vehicles to make a decision.

[1] EPA Federal Test Procedure. This is a series of tests defined by the US Environmental Protection Agency (EPA) modeling the speed of a vehicle under urban conditions.

Electric Vehicle Behaviour. The agent receives the data associated to the vehicle. It calculates its criticalities (Cr_{SoE} and Cr_{DoD}). It also receives the message from the charging station agent containing the cooperation request and the thresholds.

This agent has three possible actions: recharge the vehicle, discharge it or be idle. According to the received request, they may have several possibilities.

If it receives an *ABSORB* request, it must absorb power to cooperate. This action is in favour of the goal on the minimum state of energy. If the objective on the depth of discharge is not considered, it is not prejudicial to cooperate.

When it receives a *PRODUCE* request, it must discharge to cooperate, which is an action that may prevent its individual goal fulfillment. Yet, it can also cooperate by delaying its recharge. It does not deliver power but does not consume it either. If its criticality is lower than both thresholds, it is not critical. It can cooperate by discharging the battery. However, if its criticality is greater that one threshold, it postpones the recharge to limit its consumption. Finally, if its criticality is greater than both thresholds, it is too critical to cooperate, so it recharges the battery.

If the objective on the depth of discharge is considered, the corresponding criticality has to be taken into account. Indeed, if the vehicle has to change the sign of the charging power to cooperate, the condition to make the decision to cooperate or not has to be done taking the maximum of criticalities on the state of energy and on the depth of discharge ($Cr_{max} = \max(Cr_{SoE}, Cr_{DoD})$) or the criticality on the depth of discharge. The behaviour of vehicles is summarised in Table 1.

Table 1. Behaviour of vehicles

Current cycle	$DoD < 0$ (discharge)		$DoD > 0$ (recharge)	
Request	Conditions	Decisions	Conditions	Decisions
PRODUCE	$Cr_{SoE} < Th(discharge)$	DISCHARGE	$Cr_{max} < Th(discharge)$	DISCHARGE
	$Th(postpone) > Cr_{SoE}$ $Cr_{SoE} > Th(discharge)$	IDLE	$Th(postpone) > Cr_{SoE}$	IDLE
	$Th(postpone) < Cr_{SoE}$ $Cr_{SoE} > 0.6$	RECHARGE	$Th(postpone) < Cr_{SoE}$ $Cr_{SoE} > 0.6$	RECHARGE
ABSORB	$Cr_{DoD} < Th(postpone)$	RECHARGE	Always true	RECHARGE
	$Cr_{DoD} > Th(postpone)$	IDLE		

4.3 Synchronisation

The synchronisation of the agents has been one of the difficulties encountered in this approach. Indeed, the simulation is executed sequentially. Therefore, from a frequency simulation point of view, all vehicles make their decisions with the same view of the situation, therefore simultaneously. Depending on the number of vehicles connected, the effects of vehicles cooperation may be greater than the power deviation that need to be compensated. For example, if there is an underproduction, some vehicles will make the decision to inject power into the network in order to reduce the energy unbalance between consumption and production. Then, the total power injected into the network by the fleet may be greater than the power gap to be compensated: there is an overproduction. Therefore, oscillations can appear.

Desynchronisation by Simulation Step Decrease. A first solution to desynchronise the vehicles is to decrease the frequency simulation step. In the current system, a one-second simulation step is performed. A life cycle is then carried out for each agent, first the charging station agents and then the vehicle agents. However, for the proposed desynchronised approach, the simulation step of frequency is inferior to one second. At each step, only a fraction of the vehicle fleet makes a decision. Thus, for each step, the number of vehicles acting will be much lower: the risks of having a too strong compensation are therefore lower. Meanwhile, the vehicles still act at every second. This solution seems consistent with the assumption of an implementation on a real system since the agents would be implemented on desynchronized calculators. However, the charging station agents have a period execution time lower than the second. This assumes that the calculator running the algorithm executes its cycle in this new period, and that the system allowing the measurement of the frequency is fast enough.

Desynchronisation by Vehicle Agents Behaviour. A second possible solution is to desynchronise the vehicles through their behaviour. In fact, for this solution, the decision to cooperate by discharging, if the conditions are satisfied, is replaced by a probability to cooperate. Thus, it avoids to have too many vehicles that begin cooperating at the same time and thus reduces the risk of overcompensation. This probability is proportional to the difference between the discharge threshold defined by the charging station agent and the criticality of the agent: the higher the network criticality is, or the lower the vehicle criticality is, the greater the probability of cooperation is.

5 Results and Analysis

This last section presents the obtained results. These results are based on irradiance data from Oahu, on the 17th February 2011, from the National Renewable Energy Laboratory [17], between 4:30 pm and 6:30 pm are presented. The system is confronted to fairly large production gaps while at the same time managing the second departure of vehicles between 5:00 pm and 6:20 pm. The state of energy of the batteries at the beginning of the simulation are random between 0.3 and 1. For each simulation, the constraints on the state of energy of the vehicles at the departure time have been verified. The number of agents in each test corresponds to the number of vehicles in the situation added to the number of charging stations (2 in these simulations). The first simulations have been realised without considering the criterion on the depth of discharge of batteries.

5.1 Frequency Regulation

Influence of the Number of Vehicles on the Frequency Regulation. This first test consists in observing the frequency obtained for different sizes of fleet. Figure 1 shows the frequency obtained over a 5 min range considering a fleet of 100 and 300 vehicles. It also presents the case where no vehicle is present: the frequency is therefore not regulated. We suppose here that the synchronous generators do not play any role in second-to-second frequency regulation, as our goal is to assess the quality of this regulation when an increasing number of electric vehicles take it in charge.

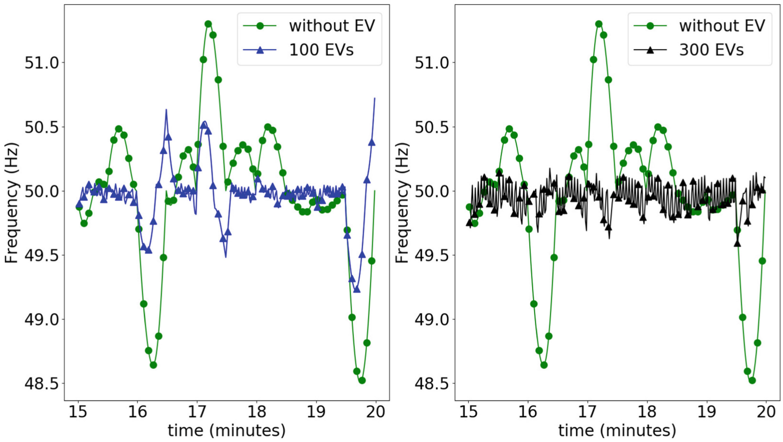


Fig. 1. Influence of the size of the fleet

These simulations show that by increasing the size of the fleet, the system reduces the fluctuations (peaks and throughs) due to the production gaps of the PV power plant more efficiently. During the simulation with 300 vehicles, the peaks are negligible. However, when the power gap is low (the frequency is stable), oscillations on the frequency appear. The amplitude of these oscillations increases with the number of connected vehicles. This phenomenon is due to the synchronisation of the vehicles present in the fleet (Sect. 4.3).

This first test shows that two properties of the system will be in conflict: the ability of the system to keep the frequency close to 50 Hz despite large production gaps and its ability to limit oscillations of the frequency.

Desynchronisation. To reduce the frequency oscillations, vehicles have to be desynchonised. Two methods have been proposed Sect. 4.3, either by reducing the simulation step or by adding a probabilistic behaviour to the vehicles.

Three cases have been simulated considering 300 EVs: reducing the simulation step (1st case), adding a probabilistic behaviour to the vehicles (2nd case) and a third one combining the two methods (3rd case). The average frequency deviation and the maximum frequency deviation are presented in Table 2.

Table 2. Results

	Synchronised vehicles			Desynchronised vehicles (300 EVs)		
	100 EVs	200 EVs	300 EVs	1st case	2nd case	3rd case
$\overline{\lvert\Delta f\rvert}$ (Hz)	0.035	0.052	0.078	0.038	0.073	0.018
$max(\lvert\Delta f\rvert)$ (Hz)	0.77	0.43	0.41	0.30	0.54	0.19

The method involving the reduction of the simulation step is more effective: the frequency oscillations are reduced. Adapting the behaviour of agents associated with EV to have a stochastic behaviour has a more reduced effect. However, when these two methods are combined, the oscillations amplitude and the maximum deviation are significantly reduced.

5.2 Limitation of Micro-cycles

For this simulation, the EV agents take into account the depth of discharge by making their decisions in a way to reduce micro-cycles of charge and discharge. Table 3 shows that the majority of the cycles experienced during the previous simulations (between 94% and 100%) have an amplitude lower than 0.09%, which corresponds to recharges or discharges of 5 s at nominal power.

Taking into account the criterion on depth of discharge and desynchronising by the two methods previously presented, only 3% of cycles have their depth of discharge under 0.09% (5 s recharge or discharge at nominal power) and 50% greater than a 30 s-long recharge or discharge.

Nevertheless, the maximum frequency deviation obtained in this simulation is 0.30 Hz and the average frequency deviation is 0.034 Hz. Thus, the frequency regulation has been lightly deteriorated compared to the previous simulations.

Furthermore, the average execution time of a cycle of the AMAS in this simulation case is 64.5 µs on a core i7-7820 HQ, 2.90 GHz of CPU, 32 Go of RAM, running on Windows 10.

Table 3. Distribution of depth of discharge of cycles

Cycles duration time (d_{cycle})	$0 < d_{cycle} \leq 5\,\mathrm{s}$	$5\,\mathrm{s} < d_{cycle} \leq 30\,\mathrm{s}$	$30\,\mathrm{s} < d_{cycle}$
Synchronised vehicles	>99%	<1%	0
Diminution of the step simulation	>99%	<1%	<0.1%
Combined methods	94%	5.7%	<0.1%
criterion on DoD taken into account	3%	47%	50%

Finally, the frequency obtained is poorly dependent on the initial state defined by the initial state of energy and the departure time of each vehicle. In the test case taking into account the depth of discharge, the maximum standard deviation on the frequency obtained for 40 simulations is 0.1556 Hz, and the average standard deviation is 0.0186 Hz.

6 Conclusion

An adaptive multi-agent system (AMAS) has been proposed to achieve the frequency regulation using electric vehicles to counterbalance the production gaps of a PV power plant between its forecast average production over 30 s and its actual production. Agents have been defined to manage at least one of the objective or constraint of the issue: frequency regulation for the charging station agents and the constraints on the batteries for the vehicle agents: minimum state of energy and cycles depth of discharge.

Agents associated with charging stations must enforce the constraint on frequency sending requests to connected vehicles for them to produce or absorb power. Each agent associated with a vehicle is then able to decide if it accepts or declines the request, according to its criticalities, and the information transmitted by the charging station agent: the cooperation request and the criticality thresholds.

The implemented system has been able to reduce the impact of the production gaps, and thus to ensure the frequency constraint, while satisfying as well the constraints on the batteries state of energy. However, the obtained frequency by this regulation oscillates significantly. These oscillations have been reduced by desynchronising the vehicles in two ways: by modifying the simulation in the way that the vehicles act at different times and by making the behaviour of the vehicles stochastic. The architecture used implies that EV agents do not communicate with each other: they only communicate with charging station agents. Moreover, the life cycle of each agent is only executed once per decision.

Other constraints of the electric grid may be taken into account in the system. For example, taking into account local network phenomena, such as voltage and line congestion, is currently investigated. In fact, this constraint would be interesting since the power circulating in each line is a local information since it can be measured only on the considered line, contrary to the frequency which is a global information in the electric grid.

Acknowledgments. The authors wish to acknoweldge Riadh Zorgati, Senior Research Engineer at EDF R&D, France, for his technical help and Alexandre Perles, from IRIT, for the AMAK framework used to support our experiments.

References

1. Haessig, P., Multon, B., Ben Ahmed, H., Lascaud, S., Bondon, P.: Energy storage sizing for wind power: impact of the autocorrelation of day-ahead forecast errors. Wind Energy **18**(1), 43–57 (2015)
2. Codani, P., Perez, Y., Petit, M.: Electric vehicles as a mobile storage device. In: Handbook of Clean Energy Systems, vol. 5 (2015). Energy Storage
3. Le Goff Latimier, R., Multon, B., Ben Ahmed, H., Baraer, F., Acquitter, M.: Stochastic optimization of an electric vehicle fleet charging with uncertain photovoltaic production. In: International Conference on Renewable Energy Research and Applications, ICRERA 2015, pp. 721–726 (2015)
4. Lund, H., Kempton, W.: Integration of renewable energy into the transport and electricity sectors through V2G. Energy Policy **36**(9), 3578–3587 (2008)
5. Rigas, E.S., Member, S., Ramchurn, S.D., Bassiliades, N.: Managing electric vehicles in the smart grid using artificial intelligence : a survey. IEEE Trans. Intell. Transp. Syst. **16**(4), 1619–1635 (2015)
6. Crawley, C., Cameron, B., Selva, D.: System Architecture: Strategy and Product Development for Complex Systems, 1st edn. Prentice Hall Press, Upper Saddle River (2015)
7. Crédo Paniah, M.K., Mercier, D., Gil-Quijano, J.: Multi-agents system for the management of renewable energy sources and mass storage. The European Power Electronics and Drives Association - Workshop: DC grid for more Renewable Energy (2013)
8. Hernandez, L., Baladron, C., Aguiar, J.M., Carro, B., Sanchez-Esguevillas, A., Lloret, J., Chinarro, D., Gomez-Sanz, J., Cook, D.: A multi-agent system architecture for smart grid management and forecasting of energy demand in virtual power plants. IEEE Commun. Mag. **51**(1), 106–113 (2013)
9. Lopes, J.A.P., Almeida, P.M.R., Soares, F.J.: Using vehicle-to-grid to maximize the integration of intermittent renewable energy resources in islanded electric grids. In: International Conference on Clean Electrical Power, ICCEP 2009, pp. 290–295 (2009)
10. Mu, Y., Wu, J., Ekanayake, J., Jenkins, N., Jia, H.: Primary frequency response from electric vehicles in the Great Britain power system. IEEE Trans. Smart Grid **4**(2), 1142–1150 (2013)
11. Gao, S., Chau, K.T., Liu, C., Wu, D., Chan, C.C.: Integrated energy management of plug-in electric vehicles in power grid with renewables. IEEE Trans. Veh. Technol. **63**(7), 3019–3027 (2014)
12. Depoorter, S., Assimon, P. M.: Les véhicules électriques en perspective, analyse coûts-avantages et demande potentielle (2011)

13. Serugendo, G.D.M., Gleizes, M.P., Karageorgos, A. (eds.): Self-Organising Software: From Natural to Artificial Adaptation. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-17348-6
14. Bernon, C., Gleizes, M.-P., Peyruqueou, S., Picard, G.: ADELFE: a methodology for adaptive multi-agent systems engineering. In: Petta, P., Tolksdorf, R., Zambonelli, F. (eds.) ESAW 2002. LNCS (LNAI), vol. 2577, pp. 156–169. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39173-8_12
15. Jorquera, T.: An adaptive multi-agent system for self-organizing continuous optimization. Doctoral thesis, Université Paul Sabatier (2013)
16. Georgé, J.P., Gleizes, M.P., Camps, V.: Cooperation. In: Di Marzo Serugendo, G., Gleizes, M.P., Karageorgos, A. (eds.) Self-Organising Software: From Natural to Artificial Adaptation. Springer-Verlag, Natural Computing Series (2011). https://doi.org/10.1007/978-3-642-17348-6
17. Sengupta, M., Andreas, A.: Oahu Solar Measurement Grid (1-Year Archive): 1-Second Solar Irradiance; Oahu, Hawaii (Data); NREL Report No. DA-5500-56506 (2010). https://doi.org/10.5439/1052451

Using Run-Time Biofeedback During Virtual Agent-Based Aggression De-escalation Training

Romy A. M. Blankendaal(✉) and Tibor Bosse

Department of Computer Science, Vrije Universiteit Amsterdam, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{r.a.m.blankendaal,t.bosse}@vu.nl

Abstract. This exploratory study discusses how biofeedback can be displayed and used in the context of virtual agent-based social skills training. Using Virtual Reality, a prototype of an aggression de-escalation training application has been developed. The application simulates an adaptive scenario in which the user interacts with a threatening virtual conversation partner. The stress level of the user, which is displayed at run-time, influences the development of the scenario. A pilot experiment is conducted to test the impact of the application on users' subjective experience (measured through a questionnaire) and physiological response (measured through skin conductance). Although preliminary, the results seem to indicate that biofeedback potentially contributes to enhancing the user's awareness of his/her own emotional state. Implications for the use of biofeedback within virtual training applications are discussed.

Keywords: Human-agent interaction · Biofeedback
Virtual reality training · Stress · Skin conductance

1 Introduction

Virtual Reality (VR) is a technique that allows people, with the use of a head mounted display, to view a 3D virtual environment in such a way that it feels like you are present inside the environment. This technology makes a wide variety of scenarios available at a low cost, without the need to travel. The immersive experience that is realized by virtual reality is used for numerous purposes, including training, therapy and entertainment.

The current paper focuses on the use of Virtual Reality for training purposes, which is an interesting area for several reasons. VR provides a high level of immersion, the environment is easy to control and to adjust, and the simulation can be interactive (by using controls). Indeed, VR has already been used, for example, to train emergency personnel for complex collaborative tasks [2] and to overcome fears such as public speaking[1].

[1] pages.samsung.com/ca/launchingpeople/English.

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 97–109, 2018.
https://doi.org/10.1007/978-3-319-94580-4_8

The focus of this research is on virtual training of social skills, which is realized by allowing the user to interact with an intelligent virtual agent. More specifically, we are interested in training people to cope with a threatening conversation partner. This is triggered by an ongoing project that aims to develop an intelligent training system for public transport employees, enabling them to practice how to de-escalate violent behaviour during confrontations with aggressive passengers [3].

During a training for threatening social circumstances, it is important for the user to learn to control his/her emotions. We hypothesize that this process can be facilitated by providing users more insight into their emotional state during training, thereby allowing them to regulate their emotions. Giving users information on their own state using physiological measurements (e.g., of their heart rate or skin conductance) is called *biofeedback*. In this paper, we explore the usefulness of biofeedback in the context of aggression de-escalation training. Two aspects of biofeedback are explore, namely (1) directly displaying the user's physiological state in the virtual environment and (2) changing the development of the virtual scenario based on the physiological measurements. Hence, the following research question is addressed: *To what extent can biofeedback help make a user aware of his/her own emotional state during interaction with a threatening agent in a VR environment?* This question is tackled by developing a prototype of a VR scenario involving a threatening virtual agent, and testing the impact of run-time biofeedback on the user's experience and stress level.

Section 2 discusses the related literature. Section 3 describes the VR prototype and Sect. 4 the architecture of the overall system, including biofeedback. Section 5 discussed the pilot study conducted to make a preliminary evaluation of the system and Sect. 6 presents the results. Finally, Sect. 7 concludes the paper with a discussion.

2 Related Literature

Virtual reality and virtual agents are used for various purposes, including stress management training for soldiers [15], exposure therapy [13], emergency training for police personnel [2], and social skill training [7]. Reasons for using virtual (reality) applications include that it enables training for dangerous situations, it is often less expensive than real-life training, and multiple training scenarios can be easily created. In addition, training scenarios can be adapted to the user's needs. However, it can be hard to create an environment which is experienced as realistic by the users. In addition, it is important for VR training applications that the skills learned in the VR training can be transferred to real-life situations.

There are a number of VR applications that make use of biofeedback. The biofeedback shown in these applications is based on measurements like heart rate[2] and breathing[3] and most of the applications are focused on trying to relax the user. The biofeedback is usually displayed as a number (such as a score), or

[2] guidedmeditationvr.com.
[3] www.exploredeep.com.

as a way to provide the user with feedback regarding the simulation (during the simulation or presented as a summary after the simulation), or used to change the virtual environment.

Changes in the virtual environment can occur in different ways. In addition to simply showing the user information about his/her physiological state, biofeedback can be used as a 'controller' for the user to interact with the virtual environment (see footnote 3), to actually change the virtual environment[4], or to trigger other subtle changes like adapting the volume [14].

When showing biofeedback, it is important that this is displayed in a clear and simple way. As the research in the intersection of VR and biofeedback currently is in its infancy, it is useful to study the design of smartphone applications. Multiple designs of smartphone applications show that the stress level of the user is often displayed as a simple meter[5], a graph[6], or a gauge[7]. In addition, the user's relative stress level is often shown as a percentage, of some maximal level and a color range is used to indicate the height of the stress level. Several mobile applications show biofeedback with the purpose of 'relaxation', i.e., to calm down the user [8]. However, our application initially tries to bring the user in a state of stress, by introducing a threatening virtual agent. This is in line with research showing that users could become more tense by interacting with agents with a negative attitude [11]. Similarly, facing a more demanding conversation agent can result in a higher experienced stress level and feeling less comfortable compared to a more understanding agent [9].

In addition to triggering a stress response in the user, we want our agent to be able to interact with the user. Realizing a conversation between a human and a virtual agent can be done in multiple ways. For instance, a dialog can be created using pre-scripted and prerecorded sentences that are triggered by the user's choices in a multiple choice menu via a *conversation tree* [3]. A disadvantage of this approach is that it makes the scenarios more static and predictable. To prevent this, various alternatives have been proposed, such as incorporating cognitive models to endow agents with a richer repertoire of behaviours [4].

To make the agent even more responsive to the user's behaviour, techniques like voice and facial expression recognition [1] or speech recognition and natural language understanding [12] could be used. This information could be used as input by the system to determine the next sentence and/or corresponding behavior of the agent as response during a conversation. Another interesting option is to allow the agent to physically 'touch' the user with the use of haptic feedback [10].

As our main purpose is to explore the potential of biofeedback during human-agent interaction, the current paper does not fully exploit such sophisticated techniques. Instead, as explained in the next sections, we allow the user to interact via free speech while using pre-recorded animations to generate the behaviour

4 themill.com/portfolio/3409/strata-?q=strata.
5 www.eco-fusion.com/serenita.
6 biofeedbackinternational.com/esense.
7 play.google.com/store/apps/details?id=com.azumio.android.stresscheck&hl=en.

of the virtual agent. The choice for which animation is selected is based on the user's physiological state.

3 Prototype Virtual Environment

The prototype developed is a VR environment in which the user has a conversation with an aggressive virtual agent. During the scenario, the goal of the user is to remain calm while talking to the agent. To remind the user to stay calm, biofeedback is used. The skin conductance of the user is measured during the scenario, providing a real-time indication of the stress level which is shown in the environment. This way, biofeedback is used as a trigger to remind the user of his/her own current stress level. In addition, the dialog of the virtual agent is adaptive, meaning that the dialog and the animations of the agent change based on the stress level of the user. When the user stays calm, the agent will become calmer, and vice versa. The conversation with the agent will take place in the target group's native language (Dutch) to make the conversation more natural.

3.1 The Scenario

The user is placed in a virtual bar, and is told to be waiting for friends (see Fig. 4). However, at some point an aggressive virtual agent walks towards the user and starts to talk. The agent starts the conversation by saying 'Hey, what are you looking at?' with an angry voice. After that, the conversation continues in a turn-based manner, where the participant can interact with the agent using free, open-ended speech.

The virtual agent makes use of an adaptive dialog based on pre-recorded sentences to talk back to the user. The dialog of the agent is a six-layer conversation tree, where each branch is a different track, representing the state of the virtual agent: being angry, neutral or calm. See Fig. 1 for a simplified view. For each step in the conversation, the next sentence is chosen from one of the three tracks. Blue, for a calm sentence if the stress level of the user was low, orange for neutral and red, and angry sentence, if the stress level of the user was high. The algorithm used to determine in which category the user's stress level falls is explained in Sect. 4.2.

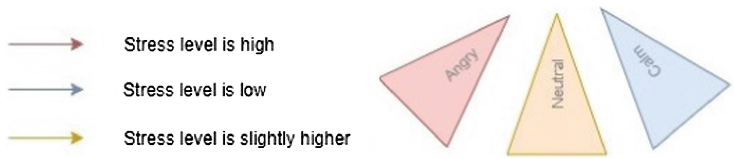


Fig. 1. The three tracks in the dialog (Color figure online)

During the conversation, the user receives real-time biofeedback regarding his/her own stress level by a display shown in the environment. The scenario always ends with the agent walking away. However, there are three different possible outcomes to the scenario (positive, neutral and negative), which provides the user with general feedback about how the conversation went.

3.2 Virtual Environment

The virtual environment is developed using the game engine software Unity[8]. The training scenario for the prototype takes place in a bar, reused from previous research [10]. Several adjustments to this environment were made, mainly to display the biofeedback.

It is important that the user is able to read his/her stress level quickly during the scenario, while it is not too distracting. The display should be simple to read and easy to understand. To achieve this, multiple sample displays were developed, and two small evaluations were conducted with four and three participants. Eventually, a *graph* and a *gauge* were evaluated as most user friendly (see Fig. 2). However, users were not unanimous in their preference for either the graph or the gauge. For that reason, both displays are used during the experiment.

Graph display Gauge display

Fig. 2. Final displays (Color figure online)

To record the sentences spoken by the aggressive agent, Audacity[9] was used. All agents in the environment and their animations were developed using the iClone Pipeline software (version 6) from Reallusion[10]. There are two kinds of agents: figures placed in the background of the environment, and the main agent that is behaving aggressively to the user (see Fig. 3). The background figures only have standard animations from the software to make them look alive. For the main agent the lips, face and body are animated using both standard and custom animations (based on the animations developed within [10]). Furthermore, with the use of C# scripts in Unity, the non-interactive agents were programmed to play the animations and speech at a specific moment within the scenario.

8 unity3d.com.
9 audacityteam.org.
10 reallusion.com.

4 System Architecture and Data Interpretation

4.1 System

The system architecture can be seen in Fig. 3. To provide the virtual agent with information on when the user is speaking or not, a microphone is used. This way, the agent is able to monitor when the user stops speaking. It will start speaking 2 s after the user stops speaking, or after the user has been speaking for more than 15 s. The microphone is not used for recording purposes.

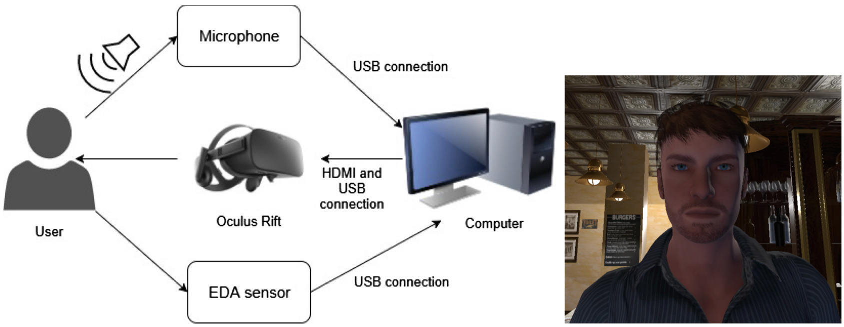


Fig. 3. System architecture and the virtual agent

The user will experience the scenario with the use of the Oculus Rift[11]. To measure the stress level of the user, skin conductance measurements are used (measuring the amount of sweat on the fingers), which are generally considered a reliable indicator for arousal [5,6]). Although other, more positively valenced emotions could in theory also lead to a higher stress level, for the prototype it is assumed that a higher skin conductance is related to a higher stress level, because the user is experiencing a threatening social situation.

Skin conductance or Electrodermal Activity (EDA) is measured via a custom-made sensor that makes use of two electrodes placed on the point and middle finger of the left hand. A Phidget I/O board was used to record the conductance and communicate with the virtual environment. Code in the virtual environment calculated the skin conductance in microSiemens values.

Multiple circumstances could have an influence on the skin conductance, like hot weather. Therefore all evaluations are conducted in the same room (with more the less the same temperature). In order to reduce noise, the participants wear the sensor on the hand they are not using, and are being asked not to move this hand.

[11] www.oculus.com/rift.

4.2 Data Interpretation

As people show large individual differences in their skin conductance level in rest, the measured values of the EDA sensor are taken relative to a user's individual baseline. The baseline value is determined by measuring the skin conductance of the person for two minutes in a state of rest, after which the average value is calculated. During the evaluation of the prototype, the baseline of each participant is measured while the user is in a calm environment, before starting the actual scenario. The sensor data is used for three different purposes:

1. Log files: To determine the baseline, the sensor data is stored, each second, in a log file together with a time stamp, over a two minute period. In addition, one log file is created for each scenario where the time, sensor data and the action of the agent is stored. This file is used for the evaluation of the experiment: to see what the stress level was of the participant during the training scenario.

2. Biofeedback: The sensor data is also used within the virtual environment to provide the participant with biofeedback regarding his/her stress level. The displays, shown in Fig. 2, are updated every second. Since there is no fixed baseline across all participants, the way the colors are displayed is personalized, dependent on the individual baseline value of a specific user. The middle of the green area represents the baseline value. The blue color represents a really low stress value. The orange and the red area, delineated through the horizontal black lines, represent situations that lead to more aggressive responses by the virtual agent. The gauge uses similar color coding. However, for pragmatic reasons, displaying the color range within the gauge was not made fully adaptive, but instead six different images were created whereby in each image the color range was depicted differently based on the baseline of the user.

3. Adaptive dialog: Finally, the data is used to make the conservation with the virtual agent adaptive. As mentioned earlier, the dialog of the agent consists of three tracks: angry (red), neutral (orange) and calm (blue). Based on the stress level of the user, the track is determined from which the next sentence is selected. The three tracks shown in Fig. 1 (red, orange, blue) thus correspond to the colored areas in Fig. 2 (red, orange, green+blue). Hence, the user's stress level directly determines the agent's reaction. To calculate the 'current' stress level, the average over the last five seconds is taken every second. To distinguish the three tracks, two thresholds are used: one threshold to separate the blue and orange tracks (th1), and one to separate the orange and the red tracks (th2). During a test session in which fluctuations in people's stress levels were measured over time, the thresholds of 0.5 and 2.0 were determined, respectively, for th1 and th2. This means, for example, that the use enters the orange track if his/her stress level is more than 0.5 (but less than 2.0) microSiemens above the baseline level. However, after a number of people had participated in the experiment, it turned out that most of the time, the agent chose sentences from the red track. Hence, to allow for more variation in the agent's behaviour (and to give the participants better possibilities to 'calm down' the agent), eventually the values of 2.0 and 3.0 were chosen.

5 Pilot Study

Because of the exploratory nature of this research, we started by conducting a pilot study, rather than a fully elaborated experiment. The goal of this study is not to explicitly test hypotheses, but to evaluate if our system works as expected and to obtain more insight in how (different types of) biofeedback is experienced during a VR scenario involving a threatening agent.

5.1 Participants

In total, 14 people participated in the study. However, two participants were excluded from the analysis due to complications during the experiment, resulting in 12 actual participants. As mentioned above, the thresholds used for the adaptive dialog were changed half way the evaluation. As a result, we ended up with two groups of 6 participants to which we will refer as group 1 and group 2. Group 1 (with th1 = 0.5 and th2 = 2.0) included 2 men and 4 women, and group 2 (with th1 = 2.0 and th2 = 3.0) included 4 men and 2 women. All participants were academic students with an age of 18 years and older (average age: 23) and had previous experience with VR environments. Due to the small number of participants, no statistical analyses were conducted.

5.2 Experimental Design

Within the experiment, people experienced the prototype VR scenario while two types of measurements were taken: subjective measurements in the form of a questionnaire, and objective measurements in the form of logs of the EDA data. The prototype containing the training scenario was shown three times to each participant: one time without biofeedback, one time with biofeedback displaying the stress level of the participant through a graph, and one time displayed through a gauge. To avoid any potential habituation effect, the order in which each participant saw the scenarios was counterbalanced.

5.3 Materials

Section 4 discussed the systems architecture. In order to use the Oculus Rift and play a high-quality VR environment, a strong gaming computer with a high-end graphics card was necessary to prevent delays during the VR experience and to create a smooth and optimal VR environment. The computer used contained an Intel i7-6800 CPU and the Nvidia GTX-1080 graphics card. In addition, a second computer was used to display the questionnaire. To make it easier for the user to look around, turn-able office chairs were used.

5.4 Procedure

Before a participant entered the room, the experimental setup was checked. The participant started by filling out the questionnaire, which included a couple of health questions to check if the participant could participate, following by some general questions. Before the start of the application, the participants were asked if everything was clear and instructed to have a conversation with the agent by just speaking out loud (using free speech). Then the EDA sensor was attached to the hand of the participant. The participants were informed that because they were wearing a VR headset, they had the ability to look around in the environment. This was explicitly stated because the angry agent will only walk towards the participant and start the conversation, when the participant is looking at him.

The participant was told that, before starting a scenario, the sensor would calibrate for two minutes. During these two minutes, the participant would hear a calm song playing, while looking at a sky during sundown. In addition, a loading circle was shown so the participants had an idea of the duration of the calibration. After the two minutes, the scenario was started. At the start of the scenario, the user is sitting at a table in a virtual bar. Within the 'real' environment the user is sitting at a table as well, to improve the feeling of immersion. Within the environment there are multiple agents, sitting around a table, standing at the bar and dancing in front of the small podium (see Fig. 4).

Fig. 4. The VR environment

The duration of the conversation with the agent was approximately two minutes. Then, the participant was requested to continue with the questionnaire. Several questions were asked regarding their experience with the virtual agent, most of which were taken from standard User Experience questionnaires (not included because of space limitations). For the scenarios with the graph and the gauge, additional questions were asked about the users' opinion on the display for the biofeedback.

After the participant watched all three scenarios, the participant filled out the final part of the questionnaire. The participants were asked some general questions about the adaptive dialog with the agent, how the training scenario was experienced, how the interactions with the virtual environment were experienced and how the participants experienced seeing their own stress level during the training scenario. The entire experiment took around 30 min.

6 Results

Due to the small sample size of this pilot study, no statistical analysis could be performed. However, several interesting qualitative results were obtained, which are summarized below.

First of all, regarding the subjective measurements, the threatening agent was experienced as aggressive by most of the participants. Additionally, compared to previous studies we conducted in this area, they reported a relatively high feeling of presence and immersion within the virtual environment. Nevertheless, the extent to which they felt physically threatened by the agent was limited.

Furthermore, the changes in the dialog turned out to behave as expected, hence validating the internal working of the system. Indeed, the questionnaires confirmed that participants did experience some changes within the aggressive level of the dialog. In group 1, more participants ended up in the red track (resulting in a more aggressive agent) than in group 2, where the agent produced more sentences from the blue track (resulting in a calmer agent). Interestingly, these differences could also be observed in the participants' reports on their experience in calming down the agent: group 1 did not really have the idea that is was possible to calm down the agent, while group 2 thought it was possible. This could well be due to the different thresholds: because group 2 did view a calmer agent more frequently, the participants may have had the feeling that they were better able to calm down the agent.

Regarding the biofeedback displays, the participants indicated that, although at times they found them somewhat distracting, they were still focused on the conversation with the agent. Seeing a display with the stress level was definitely preferred over seeing no display at all, which is a useful finding for this type of applications.

Moreover, when looking at logs of the actions selected by the agent, it seems that it was often difficult for the participants to control their stress level in order to get the agent in another track. In about half of the scenarios, the agent stayed within the same track during the entire conversation. For the other cases, some participants managed to bring the agent in a 'calmer' track (e.g., from orange to blue), but others brought the agent in an even more aggressive track (e.g., from orange to red).

Finally, it is interesting to have a close look at the objective data provided by the skin conductance measurements. Figure 5 shows the average skin conductance (relative to the baseline) over all 12 participants, plotted over time, for the three different types of displays. Here, the six numbers refer to the six sentences spoken by the agent during the conversation. Although the sample size was to small to test for statistical significance, two interesting trends can be observed. First, on average the participants' level of arousal increased during the interactions with the virtual agent. This happened in all conditions, even in the case without biofeedback display. And second, the increase in arousal seems to be higher for the conditions in which participants saw their stress level than in the condition without display. This is an interesting finding suggesting that biofeedback (possibly combined with adaptive dialogues) could be a useful method for stress induction, although this hypothesis would need to be further tested.

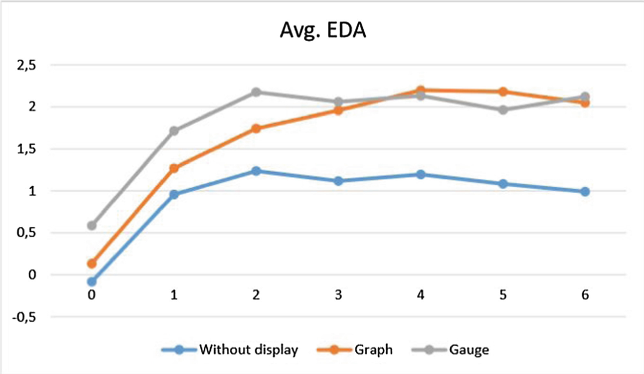


Fig. 5. Average EDA measurements for each scenario (Color figure online)

7 Conclusion and Discussion

Within this exploratory study, we investigated whether and how biofeedback can contribute to a user's awareness of his/her own emotional state during an interaction with a threatening virtual agent in a VR setting. The prototype developed for this project is a VR aggression de-escalation simulation where the user has a conversation with an angry virtual agent with the goal to calm down the agent. During the simulation, the stress level of the user (measured with an EDA sensor) is shown as biofeedback through a gauge or a graph, in order to create awareness of his/her emotional state. In addition, the stress level of the user is used as input for the agent in order to make the dialog adaptive. A pilot study was conducted, which resulted in several useful findings, including the fact that showing the participants' stress levels explicitly seemed to make them more stressed. Nevertheless, the experiment had a number of limitations, which could be addressed in follow-up studies.

An obvious limitation is the fact that the pilot experiment was based on a small number of participants. Although some first evaluations have been conducted, a more extensive experiment would be necessary. In addition, the background from the participants was the same, all participants were university students. Future research should include a more diverse user group.

The experiment also pointed out that it was often difficult for the participants to control their stress level in order to get the agent in another track. This was probably due to the thresholds used in the algorithm to generate the agent's behaviour. Besides that, especially the participants in group 1 were often not able to calm down the agent during the scenario. So, future work could pay attention to finding the ideal thresholds for adaptive dialogues, or even make the thresholds vary per individual. Also, instead of skin conductance, which is known to have a slow decay, other physiological measurements like heart rate or EEG could be used.

In addition, the scenario was rather short and simple, causing less time for the participants to adapt to the situation and experience a variety of behaviours by the agent. Moreover, the participants were explained in the beginning that they would view the scenario three times, giving them the opportunity to view how the agent changes. So, although the goal was to calm down the agent, users could ignore this to view the more angrier agent. Future research could include testing if users could learn to become calm during a longer scenario, or after using the training simulation multiple times.

The biofeedback displays were experienced as a bit distracting, although the participants were still focused on the conversation. Overall, seeing the current stress level during a VR training simulation was preferred over not seeing a display at all. So, having biofeedback was experienced as useful. There is a small preference for the gauge, but participants mention advantages for both displays. However, the design of the displays can be improved.

Finally, when looking at the skin conductance measurements, whether the user viewed their stress level by the gauge of the graph, seeing biofeedback seemed to result in a higher stress level compared to seeing no biofeedback. A possible explanation could be, that being in a stressful environment and seeing that you become more stressed makes it harder to calm down and results instead in becoming more stressed. However, additional experiments are required to confirm this hypothesis.

In conclusion, this exploratory study has resulted in some useful insights with respect to the usefulness of biofeedback in the context of human-agent interaction under threatening social circumstances. We believe that these insights have led to several pointers for further research, e.g., regarding the choice for displays and the use of skin conductance data. Such pointers are an important step towards the design of virtual training environments for training of social skills such as aggression de-escalation.

References

1. Anderson, K., et al.: The TARDIS framework: intelligent virtual agents for social coaching in job interviews. In: Reidsma, D., Katayose, H., Nijholt, A. (eds.) ACE 2013. LNCS, vol. 8253, pp. 476–491. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-03161-3_35
2. Bertram, J., Moskaliuk, J., Cress, U.: Virtual training: making reality work? Comput. Hum. Behav. **43**, 284–292 (2015)
3. Bosse, T., Gerritsen, C., de Man, J.: An intelligent system for aggression de-escalation training. In: ECAI, pp. 1805–1811 (2016)

4. Bosse, T., Provoost, S.: Integrating conversation trees and cognitive models within an ECA for aggression de-escalation training. In: Chen, Q., Torroni, P., Villata, S., Hsu, J., Omicini, A. (eds.) PRIMA 2015. LNCS (LNAI), vol. 9387, pp. 650–659. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25524-8_48
5. Boucsein, W.: Electrodermal Activity. Springer, Boston (2012). https://doi.org/10.1007/978-1-4614-1126-0
6. Chittaro, L., Sioni, R.: Affective computing vs. affective placebo: study of a biofeedback-controlled game for relaxation training. Int. J. Hum.-Comput. Stud. **72**(8), 663–673 (2014)
7. Chollet, M., Stefanov, K., Prendinger, H., Scherer, S.: Public speaking training with a multimodal interactive virtual audience framework. In: Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, pp. 367–368. ACM (2015)
8. Dillon, A., Kelly, M., Robertson, I.H., Robertson, D.A.: Smartphone applications utilizing biofeedback can aid stress reduction. Front. Psychol. **7**, 832 (2016)
9. Gebhard, P., Baur, T., Damian, I., Mehlmann, G., Wagner, J., André, E.: Exploring interaction strategies for virtual characters to induce stress in simulated job interviews. In: Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems, pp. 661–668. International Foundation for Autonomous Agents and Multiagent Systems (2014)
10. Goedschalk, L., Bosse, T., Otte, M.: Get your virtual hands off me! – developing threatening IVAs using haptic feedback. In: Verheij, B., Wiering, M. (eds.) BNAIC 2017. CCIS, vol. 823, pp. 61–75. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76892-2_5
11. Hartanto, D., Kampmann, I.L., Morina, N., Emmelkamp, P.G., Neerincx, M.A., Brinkman, W.P.: Controlling social stress in virtual reality environments. PLoS One **9**(3), e92804 (2014)
12. Kenny, P., Hartholt, A., Gratch, J., Swartout, W., Traum, D., Marsella, S., Piepol, D.: Building interactive virtual humans for training environments. In: Proceedings of I/ITSEC, vol. 174 (2007)
13. Krijn, M., Emmelkamp, P.M., Olafsson, R.P., Biemond, R.: Virtual reality exposure therapy of anxiety disorders: a review. Clin. Psychol. Rev. **24**(3), 259–281 (2004)
14. Muñoz, J.E., Paulino, T., Vasanth, H., Baras, K.: PhysioVR: a novel mobile virtual reality framework for physiological computing. In: 2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom), pp. 1–6. IEEE (2016)
15. Pallavicini, F., Argenton, L., Toniazzi, N., Aceti, L., Mantovani, F.: Virtual reality applications for stress management training in the military. Aerosp. Med. Hum. Perform. **87**(12), 1021–1030 (2016)

Check for updates

Multi-Agent Systems and Blockchain: Results from a Systematic Literature Review

Davide Calvaresi[1,2](✉), Alevtina Dubovitskaya[1,3], Jean Paul Calbimonte[1], Kuldar Taveter[4], and Michael Schumacher[1]

[1] University of Applied Sciences and Arts Western Switzerland, Sierre, Switzerland
{davide.calvaresi,alevtina.dubovitskaya,jeanpaul.calbimonte, michael.schumacher}@hevs.ch
[2] Scuola Superiore Sant'Anna, Pisa, Italy
[3] Swiss Federal Institute of Technology, Lausanne, Switzerland
[4] Tallinn University of Technology, Tallinn, Estonia
kuldar.taveter@ttu.ee

Abstract. Multi-Agent Systems (MAS) technology is widely used for the development of intelligent distributed systems that manage sensitive data (e.g., ambient assisted living, healthcare, energy trading). To foster accountability and trusted interactions, recent trends advocate the use of blockchain technologies (BCT) for MAS. Although most of these approaches have only started exploring the topic, there is an impending need for establishing a research road-map, as well as identifying scientific and technological challenges in this scope. As a first necessary step towards this goal, this paper presents a systematic literature review of studies involving MAS and BCT as reconciling solutions. Aiming at providing a comprehensive overview of their application domains, we analyze motivations, assumptions, requirements, strengths, and limitations presented in the current state of the art. Moreover, discussing the future challenges, we introduce our vision on how MAS and BCT could be combined in different application scenarios.

Keywords: Multi-Agent Systems · Blockchain
Systematic literature review

1 Introduction

Technological revolutions have deeply changed habits and customs in contemporary society. Everyday items such as smartphones, cars, clothes and household appliances are gaining increasingly sophisticated computing and communication capacities, becoming irrevocably present in everyday life. In domains such as e-health, assisted living, tele-rehabilitation, manufacturing, zero-energy buildings,

D. Calvaresi and A. Dubovitskaya—These authors share an equal contribution.

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 110–126, 2018.
https://doi.org/10.1007/978-3-319-94580-4_9

near-zero automotive fatalities, ubiquitous computing is dramatically rising, thus demanding the scientific research to push towards devices that autonomously collaborate/compete with each other [1,2]. Most of these decentralized systems implement a sort of distributed intelligence, in many cases emulating humankind dynamics. For example, in the last decades, Multi-Agent Systems (MAS) gained a crucial role in the development of intelligent distributed systems, often exchanging sensitive data [3]. In this context, accountability and trusted interactions among agents have become mandatory aspects, which entail a number of technical and scientific challenges. Despite many attempts and previous works aiming at developing models and mechanisms to guarantee communications security and trust in MAS [4–6], such requirements have not been fully satisfied yet.

Recent trends [7–10] nourish the promising idea of integrating MAS and blockchain technologies (BCT) [11,12], with the expectation of providing BCT features in use-cases where agent systems require them. However, employing a new technology such as blockchain "as-is" and by itself in dynamic and quickly evolving scenarios can represent an unlucky choice. This may be due to several reasons, spanning from fundamental properties of BCT, to application/domain specific constraints. As an example, consider the modification of blockchain code, which can happen through majority consensus. Reaching consensus in a distributed multi-stakeholder network with possible unaligned interests can be considerably complex, and new issues might be introduced as a result [13]. Although effective, some strategic decisions can hinder the evolution of the technology from academic institutions to real-world problems [14].

Nevertheless, combining BCT and MAS can represent a win-win solution if properly managed: On the one hand, the adoption and adaption of BCT may fix the security limitations broadly known in MAS literature. On the other hand, BCT can also contribute with features missing in some MAS scenarios (e.g., flexibility). For example, cloud computing systems dealing with potentially "very large datasets" are going towards a process of *agentification*, exploiting the crucial support of blockchain technology [8]. Considering agents as atomic entities populating P2P communities, the design of a fair scheduling and a general protection of the whole cluster against abusive or malfunctioning nodes is currently one of the main challenges [15]. In particular, in distributed master-less systems with reputation rating across the cluster, the application of multi-level principles of cryptocurrencies has given insightful results [15]. In fact, in [14] it is demonstrated that by combining peer-to-peer networks with cryptographic algorithms a group of agents can reach an agreement on a particular state of affairs and record that agreement, without the need for a controlling authority. The combination of blockchain and MAS in any distributed-like scenario (e.g., swarm robotics [14]) can provide the necessary capabilities to make distributed entities operations more secure, autonomous, flexible and even profitable.

A number of other examples in the literature show the recent interest in using BCT to address different challenges already present in MAS, and in various application domains. The intuition behind the integration of these paradigms and their underlying technologies is mainly driven by the needs of including features such as integrity, identity management, provenance, transaction guarantees, data security, to name a few. However, many of the implications of the results of this integration remain to be assessed. Moreover, there is a need for assessing where the combination of BCT and MAS is adequate, and what are the new challenges that arise by connecting them.

The **contribution** of this work is three-fold: (i) To better understand the motivations and the relevance of the existing contributions that combine MAS and BCT, it develops a *Systematic Literature Review (SLR)* of the current state of the art, capturing the application domains, motivations, assumptions, strengths and limitations. (ii) It *analyzes the correctness and justification* of using BCT to address the requirements of MAS. (iii) It *formalizes the open challenges* of applying BCT in practice, in particular in the framework of MAS, and *provides directions for future research*.

The paper is organized as follows: Sect. 2 introduces basic concepts, Sect. 3 presents the review process and data collection, Sect. 4 organizes and describes the obtained results, Sect. 5 discusses the obtained results, lists open challenges, and details several application scenarios, for which coupling BCT and MAS would be highly beneficial. Finally, Sect. 6 concludes the paper and presents possible future works.

2 MAS and BCT: Basic Concepts

2.1 Principles of MAS

An agent can be rationalized as an autonomous entity, with an expendable knowledge, driven by self-developed or induced objectives [16]. Moreover, agents can observe the surrounding environment through a perception layer, and possibly interact with it, as well as with other agents. MAS are generally composed of loosely coupled agents interconnected and organized in a network. The degrees of cooperation among agents, the type of application, and agent interaction model generate a broad range of behaviors. These may include concepts related to knowledge (data) sharing among agents, message-passing strategies, agreement and consensus, reputation and trust among agents, voting systems, agent identity management, and many more.

Regardless distribution and dimensions, although broadly appreciated, MAS autonomy and flexibility still generate minor concerns about possible evolution in undesired behaviors of inferences and plans. Moreover, depending on the cooperative/competitive nature of the community factors such as *trust* and *reliability* are still open challenges heavily affecting the MAS pillars: *(i)* agent local scheduler, *(ii)* communication protocol, and *(iii)* negotiation protocol [3,5].

2.2 Principles of Blockchain

Blockchain is a peer-to-peer distributed ledger technology that provides a shared, immutable, and transparent append-only register of all the actions that have happened to all the participants of the network. It is secured using cryptographic primitives such as hash function, digital signature, and encryption [17]. The data in the form of transactions, digitally signed and broadcasted by the participants, are grouped into the blocks in the chronological order and timestamped. A hash function is applied to the content of the block and forms a unique block identifier, which is stored in the subsequent block. Due to the properties of the hash function, (result is deterministic and can not be reversed) it could be easily verified if the content of the block was modified by hashing the block content again and comparing it with the identifier from the subsequent block. The blockchain is replicated and maintained by every participant. With this decentralized approach there is no need for setting up a trusted centralized entity for managing the registry. A malicious attempt to tamper the information stored in the registry will be noticed by the participants, thus guaranteeing immutability of the ledger. Many blockchains can execute arbitrary tasks, typically called smart contracts[1], written in a domain-specific or a general-purpose programming language [18].

To add a new block to the ledger a *consensus* protocol is employed [19]. Based on how the identity of a participant and its right to participate in the consensus are defined within a network, one could distinguish between public and private, permissioned and permissionless blockchain systems. In a permissionless blockchain, such as Bitcoin [17] or Ethereum [20], anyone can join the network, anyone can "write" to the shared state through invoking transactions (provided transaction fees are paid for), and anyone can participate in the consensus process for determining the "valid" state. Permissionless (or "public") blockchains are coupled to a cryptocurrency and their consensus protocols, such as proof-of-work (PoW). PoW consensus protocol was presented in [17] in the framework of the first application of blockchain technology for Bitcoin cryptocurrency management. PoW is based on so-called "mining": a process of looking for a nonce – a random number that is stored in every block – so that the resulting hash of a new valid block satisfies certain requirements. These requirements set the difficulty threshold for the process of finding the nonce and determines the average number of hashes needed to mine one block. This impacts the amount of energy to be spent to find such nonce. In 2013 the amount of energy used by Bitcoin mining was already comparable to the Irish national energy consumption [21]. Existing PoW blockchains can achieve throughput of not more than 60 transactions per second without significantly affecting the blockchain's security [22]. These findings show that PoW can negatively impact the system scalability and overall throughput [23]. Trying to address these issues researchers have challenged various aspects of the Bitcoin system and proposed modifications in its core operation, e.g., modification of the block generation rate or alternative proof of work implementations. A security analysis of PoW based consensus protocols can be found in [24].

A permissioned blockchain in contrast has means to identify the nodes that can control and update the shared state, and often also have ways to control

[1] Smart contract and chaincode logic concepts are quite close, therefore, we use the former when talking about programmable contract, or a set of rules, when discussing BCT in general.

who can issue transactions. Consensus protocols for reaching an agreement by exchanging messages even if some nodes fail, collude, or send the corrupted messages could be employed in permissioned blockchain systems. In [19] the authors present an overview of consensus protocols used in the context of permissioned blockchains. The authors also review the underlying principles, and compare the resilience and trustworthiness of some protocols as well as the permissioned blockchain systems (e.g., Hyperledger Fabric, Tendermint, R3 Corda, and MultiChain).

Another approach to constructing a blockchain relies on properties of Keyless Signatures Infrastructure (KSI) [25]. KSI is a globally distributed system for providing time-stamping and server-supported digital signature services. KSI blockchain (Guardtime) employs chain-resistance property of hash functions to verify the integrity of the data using hash-chain. KSI Blockchain deployed, for instance, in Estonian government networks to insure the integrity of the data stored in government repositories and protect them against insider threats. The number of participants in the KSI consensus protocol is limited, which allows to eliminate the need for PoW and ensuring that settlement can occur within one second. However, major drawbacks of such approach is limited decentralization and required trust in the participants of KSI consensus.

3 Review Methodology

To provide a comprehensive study, we have opted for performing a systematic, rigorous and reproducible process of retrieval, selection, and analysis of relevant literature. This paper adheres to the procedure adopted and adapted by [28] (see Fig. 1).

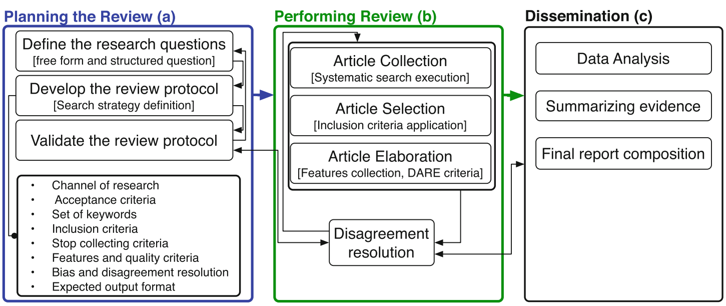


Fig. 1. Review methodology structure according to [26,27].

Following the Goal-Question-Metric (GQM) [29], the generic free-form question *"What challenges demand the employment of BCT in MAS, and is this employment justified?"* is broken-down in following *structured research questions* (SRQs).

SRQ1: *How has the combination of BCT in MAS evolved over the years in terms of **when** (year) and **where** (geographical indication of the scientific institute) such research took place?* **SRQ2:** *What are the motivations supporting the employment of blockchain in agent-based systems?* **SRQ3**: *What are the **application domains** and **scenarios** used to test or employ blockchain technologies? What are the **requirements** such approaches aimed at meeting?* **SRQ4:** *What are the **mechanisms** and related **assumptions** within the proposed protocols/approaches characterizing the most relevant contributions?* **SRQ5:** *What are the **strengths** and **limitations** that such technologies might imply?* **SRQ6:** *What are the **stated future research directions** and **challenges** identified by the scientific community?*

To perform a more accurate semi-automatic research, some keywords have been contextualized. Based on the reviewers' rooted backgrounds on MAS and BCT domains, following keywords were defined: *Blockchain + MAS, Blockchain + multi-agent system, Blockchain + smart contract + multi-agent system, smart contract + multi-agent system + trust*. The initial collection counted 36 papers. A further coarse-grained examination, processing the compliance of the selected abstracts with the following *inclusion criteria*, reduced them to 14.

(A) **Context:** The primary studies should define their contributions in the context of blockchain technology employed in agent-based systems.
(B) **Purpose:** The purpose of primary studies has to refer to applying BCT seeking for traceability, commitment, security, and trust in MAS mechanisms such as negotiation, coordination, collaboration, and competition.
(C) **Theoretical foundation:** The primary studies should provide at least one of the following elements: [visionary formulation, theoretical definition, system design].
(D) **Practical contribution:** The primary studies should provide at least one of the following elements: [practical implementation, tests, critical analysis, critical evaluations or discussion].

4 Results Presentation

This section presents the outcomes obtained by performing the methodology presented in Sect. 3. We are following the research questions presented above.

To answer **SQR1** we looked where and when the research presented in the selected papers has been conducted. Almost all the articles were published in 2017 (10 studies), while among them, one study [10] as well as another work from 2016 [14] were found on arXiv e-Print. In addition, one paper was published in 2015, and one is accepted to be presented on a congress that will take place in 2018 [36]. Due to the adoption of KSI blockchain technology in Estonia, a number of works (4 out of 14) were produced entirely or partially in Estonia. Otherwise, researchers all over the world started exploring the possibilities of employing BCT in the area of MAS: one can find contributions form Europe, US, Japan, China, and Russia.

Table 1. Domains and application scenarios.

Domain	Application	Ref.	Astr.
Collaborative governance	- transactive energy systems	[7]	T
	- conflicts resolution in business collaboration	[9]	C
	-legal accountability (regular and self-aware contracts)	[30]	P
	- task, allocation, coordination, and supervision of a group of people who share, common economic interests	[13]	C
Big data management	- management and collection of big data in highly distributed environment	[8]	C
	- anonymization of distributed data	[31]	T
Coordination	- distributed artificial intelligence	[10]	C
	- swarm robotics	[14]	C
	- coordination models in IoT	[32]	C
Trust, data integrity, reputation management	- reputation management in P2P clusters	[15]	T
	- formalisation of validation and authentication protocol for secure identity assurance	[33]	P
	- eCommerce, demand-supply relationships	[34]	P
	- supply and value chains in industrial environments and marketplaces	[35]	C
	- software life-cycle development	[36]	C

Concerning **SQR2**, we analyzed the *opportunities* and the *motivations* characterizing MAS and BCT with respect to existing technologies/approaches. The motivation for applying BCT in the framework of MAS was almost always based on the *requirements* determined by the application scenario for the multi-agent systems in question. We list the requirements with the corresponding applications when answering **SQR3** below. The need for accountability, transparency, and trust is of high importance in many applications of MAS. In [32], the authors, for instance, are seeking for establishing accountability, traceability, and transparency of interaction for tuple-based coordination. In [9,31,33,35] issues and challenges related to identification and trust are raised by the authors. Employing smart contracts, and having an access to the distributed ledger, in addition to addressing the issues just mentioned above, could also provide a simplified solution for a distributed master-less reputation management as stated in [15].

Addressing **SQR3**, we grouped selected works based on their *domains* (i.e. classification of the main purpose or usage of BCT), and listed the *application scenarios* explored. Table 1 presents the domains and the application scenarios, for which combination of MAS and BCT principles could be beneficial, as stated in the analyzed research works. For every paper, based on the maturity of the work, we specify the abstraction level: Conceptual (C), Prototype (P), or Tested (T). One could notice that more then 50% of the works are still at the conceptual level (C). Only three articles present system prototypes (P) and the three others provide evaluation of the proposed solutions.

The majority of the papers focus on collaborative governance and trust and reputation management. Among the applications, the following use-case scenarios are elaborated the most: an approach for energy trades in an open P2P market, while insuring anonymity and security of the participants [7]; a prototype of a system for blockchain-driven self-aware agents-assisted contracts [30], a distributed multi-level reputation scoring system implemented into P2P clusters [15]; trust management for both supply-demand relationships [34], and for the process of software verification [33]. In a more orthogonal position, [13] reveals a number of challenges, potential difficulties, and pitfalls that have to be considered when applying BCT.

Using blockchain technology for management of big distributed datasets and for coordination mechanisms between the agents were also proposed. The authors in [31], for instance, suggest to use BCT for keeping the log of all the transactions such as creation of privacy policy, data exchange, and data anonymization. Regarding the employment of BCT for agent coordination, only very preliminary results were found. These works just motivate the possibility to use BCT, or mention coupling MAS and BCT as a future research possibility. For instance, blockchain technology were proposed to be applied when multiple swarms from competitor companies have to coexist in the same environment, such as in mining scenarios, intelligent transportation environments, or search and rescue missions [14]. The authors in [10] try to address the drawback of existing Artificial Intelligence (AI) systems that are managed by a central server (thus, distributed, but not decentralized). Therefore, these systems are limited in performing complex organization and planning, and in solving composite tasks. The authors propose to use BCT for decentralization of AI. This would allow to solve problems with virtually unlimited power and maximum efficiency [10]. According to [32], BCT could be used to bring accountability, traceability, and transparency of interactions, and to strengthen suitability of applying the tuple-based coordination models (such as TuCSoN [37]), in different domains, such as healthcare, or IoT.

To answer **SQR4** we list the *mechanisms* and related *assumptions* characterizing the existing solutions. We focus on the papers that present solutions that were prototyped or already tested. For the transactive energy system proposed in [7] Kvaternik et al. employ a decentralized computation fabric based on Ethereum, a novel trading sequence implementation (including fulfillment of partial trades), as well as off-blockchain communications. Kiyomoto et al. [15], in the framework of multi-level scoring systems for P2P clusters, employ BigchainDB [38] – a technology that combines the properties of decentralized processing platforms like Ethereum, and decentralized file systems like InterPlanetary File System (IPFS). The authors assume that in the MAS the agents are not trusted i.e., each node can play his own game. Private BCT, Hyperledger v0.6 is used in [31] for trading anonymized datasets. ADI (anima-desire-intention) model with 6 dimensions (physiology, belief, character, knowledge, experience, and context) are employed in [34] to model interactions of supply-demand information agents.

Next question, **SQR5**, concerns the advantages of the research work presented in the selected papers. We first summarize the strengths and then list the limitations that were identified by the authors. It is broadly acknowledged that BCT enables business collaborations that require high-reliability and shared, trusted, privacy-preserving, immutable data repositories, and smart contracts execution [39]. Moreover, coupling BCT-based smart contracts with MAS opens the door to new interesting scenarios, such as simplifying the distributed governance of groups of people [13]. By doing so, transaction costs for reaching an agreement can be reduced, fostering the formalization and enforcing relationships between people, institutions, and the assets they own, by standardizing transaction rules [13]. MAS dynamics are a very close representation of human society, therefore, tracking their interactions while guaranteeing their immutability can prevent situations where two or more parties claim the opposite about whether a payment or a service has been performed. Moreover, besides immutability of values, this entails that events are immutable over time (timestamped).

The language for executing self-aware contracts (SAC) proposed in [30] is based on obligations and a more static and declarative approach. However, the relation between declarative and imperative programming in smart contracts is still an open challenge. Thus, whether logic (e.g., voting protocols/algorithms) can also be covert in this way is another interrogative. Nevertheless, there is still a lack of a framework supporting migration from a smart- towards a self-aware contract. The latter has the ability to gather information about their internal/external-contextual state and progress to reason about their behavior while being a law artifact [30]. Immutability implies storing (machine-readable and agent-executable) the contract and its obligations, which to this extent, might increase the complexity. [7] also mentions one limitation related to Solidity[2] a language for creating smart contracts on the Ethereum platform.

We finally focus on **SQR6** and summarize future challenges. As the majority of the papers present the work at the conceptual level, implementation of the proposed scenarios was stated as a future challenge by the authors in [8,10,14,32,33,35,36]. More mature solutions, for instance, [7], identify the need to improve anonymity of the nodes, in case of employing public blockchain implementation such as Ethereum. Taking into account the risks of de-identification, due to the possibility to track and link the transactions of a user [40,41], the authors in [7] suggest some countermeasures, such as onion routing [42], or employing a large number of anonymous addresses. However, in practice, these countermeasures do no guarantee that de-identification of a user is not possible. Another work [15] indicates the need to perform large-scale evaluation and the need to achieve scalability of their distributed multi-level reputation scoring system. Developing

[2] https://solidity.readthedocs.io/en/develop/.

economical models and defining optimal settings for the platform, as proposed in [31] for datasets trading, was also listed as a future challenge.

5 Discussion

When a new technology is unveiled or it makes a brake-through in new application domains, many questions arise, especially concerning when and how to apply it depending on requirements and context. The emergence of BCT and its integration in agent-based systems requires a better comprehension of the implications and the impact of this new technology, as we have seen in the research works analyzed in this paper. Hence, in this section we first aim at understanding whether combining BCT and MAS is justified, or if alternative approaches might be employed. Then, after discussing about current solutions and our vision, we identify the main open challenges in the field.

Justification of Binding MAS and BCT. Papers providing mostly conceptual contributions discuss the potential benefits of combining BCT and MAS. However, no in-depth analysis, demonstration, or concrete evaluation of its necessity are provided. For example, the authors in [8], suggest using MAS to solve the scalability issues, common in most blockchain architectures. Nevertheless, no further details are provided on how this could be concretely achieved. Moreover, in [9], the authors present smart contracts developed for cross-organizational collaboration. Then, it is only mentioned as a future direction the possibility of exploring how blockchain technology can realize non-repudiation in process-aware smart contracting governance. Given that many of these works are still on a preliminary stage, it is the case that several key aspects such as the ones above-mentioned are still undeveloped.

Correctness in Applying BCT in the Framework of MAS. Concerning the findings provided by concrete solutions, the necessity of such contribution is often questionable, and therefore it might be that the employment of BCT is also questionable. For example, the authors of [31] propose to use private blockchain as a "service" to keep track of the agreements provided by owners of data and transactions containing information about sharing anonymized datasets, in order to allow the data owners to track all the events of the data sharing process. While keeping an immutable log containing all the user agreements is very important, it does not fully justify the use of private blockchain technology. According to the existing laws on the area of management of information about individuals (including sensitive data)[3], once the data are properly anonymized such that de-identification is not possible by any reasonable means, these data do not belong to the initial data owner anymore. Thus, keeping the log about all the transactions, as well as participating in the consensus protocol (especially for

[3] EC Data Protection Directive 95/46/EC; Health Insurance Portability and Accountability Act.

the data clients that do not need to access multiple data-sets) may be a burden and will not contribute to the usability or adoption of the system.

Furthermore, it is worth to focus on the correctness of the BCT application in a given problem or scenario. However, it is challenging to justify the correctness of the solutions that are still at the conceptual stage. Nevertheless, some of the selected primary studies, describing more developed solutions, list as well their argumentation supported by the presented evaluation results [7,15]. For instance, in [31], the authors mention that one of the benefits of their approach is that no central server managed by a trusted third party (TTP) is required, therefore the cost of deployment of such system can be reduced as there is no need to maintain the TTP. However, the solution in [31] uses Hyperledger Fabric – implementation of the private blockchain technology – and therefore, requires membership service, as well as certification authority for registering users and distributing the credentials (public/private keys).

Open Challenges of BCT Towards MAS Application Scenarios. Even in cases where the use of BCT was justified and correctly employed, several challenges still need to be addressed (the following analysis extends the one provided in [13]): *(i)* Creating a legal base for BCT; *(ii)* Verifying correctness of the chaincode/smart contracts; *(iii)* Preserving the distributed nature of BCT, by preventing creation of mining pools, and collusion among the nodes in the framework of public BCT; *(iv)* Developing solutions to ensure privacy and anonymity where appropriate; *(v)* Ensuring adoption of the new blockchain technology; *(vi)* Managing membership service in the framework of permissioned BCT; *(vii)* Addressing scalability issues of BCT; *(viii)* Ensuring reliability of the mechanisms on which BCT is built and are often used in combination with e.g., key management, hashing, digital signature, encryption. Although some of these challenges are also present outside MAS scenarios, addressing appropriately will have a deep impact on the adoption of BCT in agent-based systems.

It is also highly important to evaluate whether the MAS requirements for every application scenario could be addressed without BCT. The authors in [43] provide a diagram that could be used to help to evaluate whether the use of BCT is justified. One has to take into account that given that this technology is still being developed, there is a number of issues that may affect early adopters. These include the lack of scalability of the current implementations of BCT, as well as the complexity of the blockchain technology, which could be more substantial then the benefits brought when applying it. Choosing the technology implementation (public/private BCT), defining what kind of data should be stored on-, and off-blockchain are essential questions. In the majority of the papers selected for the review, these points were not adequately addressed. Table 2 shows how the properties of BCT can fulfill multiple MAS requirements.

According to the evidence elaborated in this study, reputation, transparency, and traceability are crucial in case of competitive behavior among the agents, whereas trust and accountability are of a high importance for collaborative behavior.

Table 2. Mapping between MAS requirements and the main properties of the BCT.

MAS requirements	BCT properties				
	Immutability	Complete history	Distributed consensus	Cryptography primitives (e.g., hash, digit sign)	Smart contract
Trust	X	X	X	X	X
Reputation	X	X	X		
Data integrity	X	X	X	X	
Traciability	X	X	X		X
Transparency	X	X	X	X	X
Anonymity				X⋆	
Privacy	X			X⋆	
Authenticity	X	X	X	X	

Unfortunately, not all the features of MAS requiring support could gain advantages by applying BCT to MAS. However, Table 2 proposes a possible mapping elicited by studying current solutions or the proposed designs. For instance, the authors in [9] propose to investigate how BCT could be used to address privacy issues, yet, blockchain solely does not provide a general solution for privacy and anonymity [40,41]. However, BCT could be used for ensuring/enforcing role-based access control, or privacy-policy management [31,32,44]. To address the anonymity and privacy requirements, additional mechanisms have to be employed, e.g., crypthographic primitives used off-blockchain (hence, marked with the ⋆ in the Table 2), secret sharing scheme [45], secured multi-party computations [46] as proposed in [47], communication anonymity solution, such as onion routing, as mentioned in [7]. However, the scalability limitations of BCT, especially in case of using private blockchain technology, can create a barrier when applying BCT.

Employing MAS and BCT in Real-World Applications. Hereafter, we present our vision regarding potential applications where coupling BCT and MAS could be highly beneficial. For instance, in the healthcare domain, and particularly in connected health, the following scenario could be considered. Every actor (e.g., caregiver, insurance, pharmacy, healthcare/ambient assisted living device) can be modeled as an agent with a different behavior. Some agents could be cooperative and trusted (e.g., caregiver and pharmacy), and the others may require reputation management, and transparency in order to ensure correct behavior. Such MAS would benefit from all main properties of the blockchain: immutability, traceability, distributed consensus, use of cryptographic primitives, and ability to define functionality of the system using smart contracts. Smart contracts could be used for managing insurance claims, reimbursements of the medications, payments of medical visits, privacy policy management. Designing emergency access to the data could be based on the BCT and deployed using witness *cothority* [48]: a "collective authority" whose purpose is to witness, validate, and cosign the statements.

Moreover, BCT can be employed for data sharing, and evaluating an anonymity level of individuals given access control policy, and shared data. This approach may bear similarities with [31], yet it has an important conceptual difference: not only to track the datasets that contain the data owner information [31], but rather maintain what kind of information were exactly shared, and use this to adjust the anonymization process for the consequent data releases, or updates. Employing BCT in different healthcare scenarios has already been proposed in [49–51], and several prototypes exist [44,52]. Principles of MAS are as well often applied in healthcare domain [27,53,54]. Possibility to combine MAS and BCT to improve healthcare management has already been mentioned in [32].

MAS can be as well successfully combined with BCT in information systems for supporting business-to-business (B2B) electronic commerce, where software agents represent different companies involved in B2B e-commerce [55]. Until now, distributed systems of this kind have required a trusted mediator that stores the transactions occurring between the parties, such as ordering, supplying, and paying. As has been argued in [33], BCT enables to get rid of the role of such a mediator by storing transactions in a distributed ledger based on BCT. Moreover, in addition to transactions, also commitments and meta-commitments can be securely stored in a distributed ledger.

Another area where the combination of MAS and BCT can be useful is societal information systems [56] that gather information from hundreds of nodes, each associated with a person. Recently, such systems became known as platforms for sharing economy, such as Uber and AirBnB. These platforms usually have a central mediator that processes the information at its disposal in its own interests and only selectively shares it between the participants. Systems of this kind can be "democratized" so that they would better reflect the spirit of sharing economy by representing each node in a network by a software agent.

6 Conclusions

This paper proposed an SLR applied to 14 primary studies supporting the adoption of BCT in MAS. An overview of their domains, requirements of the application scenarios, motivations, assumptions, strengths, limitations, and identified future challenges has been provided. We also discussed correctness and justification of using BCT to address the requirements of MAS. We then proposed our vision on how MAS and BCT could be combined in different application scenarios. Ongoing work focuses on addressing open challenges of employing blockchain technology in the framework of MAS, formalizing them in a roadmap. Future work includes the implementation of the conceptual solutions that propose adequate applications of BCT in MAS.

References

1. Schatten, M., Ševa, J., Tomičić, I.: A roadmap for scalable agent organizations in the internet of everything. J. Syst. Softw. **115**, 31–41 (2016)
2. Calvaresi, D., Sernani, P., Marinoni, M., Claudi, A., Balsini, A., Dragoni, A.F., Buttazzo, G.: A framework based on real-time OS and multi-agents for intelligent autonomous robot competitions. In: IEEE Symposium on Industrial Embedded Systems (SIES), pp. 1–10 (2016)
3. Calvaresi, D., Marinoni, M., Sturm, A., Schumacher, M., Buttazzo, G.: The challenge of real-time multi-agent systems for enabling IoT and CPS. In: IEEE/WIC/ACM International Conference on Web Intelligence (2017)
4. Yu, B., Singh, M.P.: An evidential model of distributed reputation management. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1, pp. 294–301. ACM (2002)
5. Ramchurn, S.D., Huynh, D., Jennings, N.R.: Trust in multi-agent systems. Knowl. Eng. Rev. **19**(1), 1–25 (2004)
6. Hedin, Y., Moradian, E.: Security in multi-agent systems. Procedia Comput. Sci. **60**, 1604–1612 (2015). (In: Proceedings of Knowledge-Based and Intelligent Information and Engineering Systems 19th Annual Conference, KES-2015, Singapore, September 2015)
7. Kvaternik, K., Laszka, A., Walker, M., Schmidt, D., Sturm, M., Dubey, A., et al.: Privacy-preserving platform for transactive energy systems. arXiv preprint arXiv:1709.09597 (2017)
8. Qayumi, K.: Multi-agent based intelligence generation from very large datasets. In: 2015 IEEE International Conference on Cloud Engineering (IC2E), pp. 502–504. IEEE (2015)
9. Norta, A., Othman, A.B., Taveter, K.: Conflict-resolution lifecycles for governed decentralized autonomous organization collaboration. In: EGOSE, pp. 244–257 (2015)
10. Ponomarev, S., Voronkov, A.: Multi-agent systems and decentralized artificial superintelligence. arXiv preprint arXiv:1702.08529 (2017)
11. Swan, M.: Blockchain: Blueprint for a New Economy. O'Reilly Media Inc., Newton (2015)
12. Tapscott, D., Tapscott, A.: Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World. Penguin, London (2016)
13. Shermin, V.: Disrupting governance with blockchains and smart contracts. Strateg. Change **26**(5), 499–509 (2017)
14. Ferrer, E.C.: The blockchain: a new framework for robotic swarm systems. arXiv preprint arXiv:1608.00695 (2016)
15. Gattermayer, J., Tvrdik, P.: Blockchain-based multi-level scoring system for P2P clusters. In: International Conference on Parallel Processing Workshops, ICPPW, pp. 301–308. IEEE (2017)
16. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach, pp. 111–114. Prentice Hall, Upper Saddle River (2003)
17. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008)
18. Pass, R., Shi, E.: Hybrid consensus: efficient consensus in the permissionless model. In: LIPIcs-Leibniz International Proceedings in Informatics, vol. 91, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2017)
19. Cachin, C., Vukolić, M.: Blockchains consensus protocols in the wild. arXiv preprint arXiv:1707.01873 (2017)

20. Buterin, V.: Ethereum: a next-generation smart contract and decentralized application platform (2014). https://github.com/ethereum/wiki/wiki/%5BEnglish%5D-White-Paper
21. O'Dwyer, K.J., Malone, D.: Bitcoin mining and its energy footprint. In: ISSC 2014/CIICT 2014, pp. 280–285. IET (2013)
22. Gervais, A., Karame, G.O., Wüst, K., Glykantzis, V., Ritzdorf, H., Capkun, S.: On the security and performance of proof of work blockchains, pp. 3–16 (2016)
23. Vukolić, M.: The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. In: Camenisch, J., Kesdoğan, D. (eds.) iNetSec 2015. LNCS, vol. 9591, pp. 112–125. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39028-4_9
24. Kiayias, A., Panagiotakos, G.: On trees, chains and fast transactions in the blockchain. IACR Cryptology ePrint Archive (2016)
25. Buldas, A., Kroonmaa, A., Laanoja, R.: Keyless signatures' infrastructure: how to build global distributed hash-trees. In: Riis Nielson, H., Gollmann, D. (eds.) NordSec 2013. LNCS, vol. 8208, pp. 313–320. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41488-6_21
26. Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., Linkman, S.: Systematic literature reviews in software engineering - a systematic literature review. Inf. Softw. Technol. **51**(1), 7–15 (2009)
27. Calvaresi, D., Cesarini, D., Sernani, P., Marinoni, M., Dragoni, A., Sturm, A.: Exploring the ambient assisted living domain: a systematic review. J. Ambient Intell. Hum. Comput. 1–19 (2016)
28. Calvaresi, D., Appoggetti, K., Lustrissimini, L., Marinoni, M., Sernani, P., Dragoni, A.F., Schumacher, M.: Multi-agent systems' negotiation protocols for cyber-physical systems: results from a systematic literature review. In: Proceedings of ICAART (2018)
29. Kitchenham, B., Brereton, P., Turner, M., Niazi, M., Linkman, S., Pretorius, R., Budgen, D.: Refining the systematic literature review process-two participant-observer case studies. Emp. Softw. Eng. **15**(6), 618–653 (2010)
30. Norta, A., Vedeshin, A., Rand, H., Tobies, S., Rull, A., Poola, M., Rull, T.: Self-aware agent-supported contract management on blockchains for legal accountability. White paper (2017)
31. Kiyomoto, S., Rahman, M.S., Basu, A.: On blockchain-based anonymized dataset distribution platform. In: 2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA), pp. 85–92. IEEE (2017)
32. Mariani, S., Omicini, A., Ciatto, G.: Novel opportunities for tuple-based coordination: XPath, the blockchain, & stream processing (2017)
33. Leiding, B., Norta, A.: Mapping requirements specifications into a formalized blockchain-enabled authentication protocol for secured personal identity assurance. In: Dang, T.K., Wagner, R., Küng, J., Thoai, N., Takizawa, M., Neuhold, E.J. (eds.) FDSE 2017. LNCS, vol. 10646, pp. 181–196. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70004-5_13
34. Shen, J., Shen, J., Huang, Y., Huang, Y., Chai, Y., Chai, Y.: A cyber-anima-based model of material conscious information network. Int. J. Crowd Sci. **1**, 9–25 (2017)
35. Bonino, D., Vergori, P.: Agent marketplaces and deep learning in enterprises: the composition project. In: Annual Computer Software and Applications Conference, pp. 749–754 (2017)
36. Fuller, T.R., Deane, G.E.: Anomaly detection and intelligent notification. In: Future of Information and Communications Conference (2018)

37. Omicini, A., Zambonelli, F.: Coordination for internet application development. Auton. Agent. Multi-agent Syst. **2**(3), 251–269 (1999)
38. McConaghy, T., Marques, R., Müller, A., De Jonghe, D., McConaghy, T., McMullen, G., Henderson, R., Bellemare, S., Granzotto, A.: Bigchaindb: a scalable blockchain database. White paper, BigChainDB (2016)
39. Hull, R., Batra, V.S., Chen, Y.-M., Deutsch, A., Heath III, F.F.T., Vianu, V.: Towards a shared ledger business collaboration language based on data-aware processes. In: Sheng, Q.Z., Stroulia, E., Tata, S., Bhiri, S. (eds.) ICSOC 2016. LNCS, vol. 9936, pp. 18–36. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46295-0_2
40. Androulaki, E., Karame, G.O., Roeschlin, M., Scherer, T., Capkun, S.: Evaluating user privacy in bitcoin. In: International Conference on Financial Cryptography and Data Security (2013)
41. Möser, M.: Anonymity of bitcoin transactions. In: Münster Bitcoin Conference, pp. 17–18 (2013)
42. Dingledine, R., Mathewson, N., Syverson, P.: Tor: the second-generation onion router. Technical report, Naval Research Lab Washington DC (2004)
43. Wüst, K., Gervais, A.: Do you need a blockchain? IACR Cryptology ePrint Archive (2017)
44. Dubovitskaya, A., Xu, Z., Ryu, S., Schumacher, M., Wang, F.: Secure and trustable electronic medical records sharing using blockchain. arXiv preprint arXiv:1709.06528 (2017)
45. Shamir, A.: How to share a secret. Commun. ACM **11**, 612–613 (1979)
46. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: ACM Symposium on Theory of Computing (1988)
47. Zyskind, G., Nathan, O., et al.: Decentralizing privacy: using blockchain to protect personal data. In: Security and Privacy Workshops (2015)
48. Syta, E., Tamas, I., Visher, D., Wolinsky, D.I., Jovanovic, P., Gasser, L., Gailly, N., Khoffi, I., Ford, B.: Keeping authorities "honest or bust" with decentralized witness cosigning. In: Symposium on Security and Privacy, pp. 526–545, May 2016
49. Kuo, T.T., Kim, H.E., Ohno-Machado, L.: Blockchain distributed ledger technologies for biomedical and health care applications. J. Am. Med. Inform. Assoc. **24**(6), 1211–1220 (2017)
50. Yue, X., Wang, H., Jin, D., Li, M., Jiang, W.: Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. J. Med. Syst. **40**, 218 (2016)
51. Dubovitskaya, A., Xu, Z., Ryu, S., Schumacher, M., Wang, F.: How blockchain could empower eHealth: an application for radiation oncology. In: Begoli, E., Wang, F., Luo, G. (eds.) DMAH 2017. LNCS, vol. 10494, pp. 3–6. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67186-4_1
52. Ekblaw, A., Azaria, A., Halamka, J.D., Lippman, A.: A case study for blockchain in healthcare "MedRec" prototype for electronic health records and medical research data (2016)
53. Calvaresi, D., Schumacher, M., Marinoni, M., Hilfiker, R., Dragoni, A.F., Buttazzo, G.: Agent-based systems for telerehabilitation: strengths, limitations and future challenges. In: Montagna, S., Abreu, P.H., Giroux, S., Schumacher, M.I. (eds.) A2HC/AHEALTH -2017. LNCS (LNAI), vol. 10685, pp. 3–24. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70887-4_1

54. Dubovitskaya, A., Urovi, V., Barba, I., Aberer, K., Schumacher, M.I.: A multiagent system for dynamic data aggregation in medical research. BioMed. Res. Int. (2016)
55. Taveter, K.: Agile engineering of B2B automation systems. ERCIM News **50**, 60–61 (2004)
56. Taveter, K., Du, H., Huhns, M.N.: Engineering societal information systems by agent-oriented modeling. J. Ambient Intell. Smart Environ. **4**(3), 227–252 (2012)

A Model and Platform for Building Agent-Based Pervasive Mixed Reality Systems

Angelo Croatti(✉) and Alessandro Ricci

Computer Science and Engineering Department (DISI),
Alma Mater Studiorum – University of Bologna, Via Sacchi, 3, Cesena, Italy
{a.croatti,a.ricci}@unibo.it

Abstract. In agent literature, a partially unexplored area is related to the integration of ever-wider opportunities offered by technologies such as Mixed Reality (MR) and Augmented Reality (AR). In this paper we present a framework called *Augmented Worlds* (AW), which provides a model and a technological support to develop a broad spectrum of agent-based AR/MR systems. Distinguishing key features of the approach include: bi-directional augmentation, support for existing cognitive agent technologies, support for developing open multi-user environments. In the paper, we describe first the conceptual model on which the framework is based, and then a concrete architecture and prototype implementation. Two case studies about real-world applications – an augmented museum and an augmented harbour – engineered with the framework are finally discussed.

Keywords: Mixed Reality · Augmented Reality
Augmented Worlds · Agent programming technologies · BDI

1 Introduction

AR and MR are quickly becoming mainstream technologies to be exploited for designing smart environments blending physical and virtual objects, introducing new opportunities in supporting individual and cooperative human activities. Augmented Reality can be defined as a medium in which digital information is added (superimposed) to the physical world in registration with the world itself and displayed to a user dependently on its location and its perspective [1,14]. Such a registration may occur in different ways, e.g. by means of fiducial markers placed in the environment, perceived and processed through the camera(s) mounted on smart-glasses or devices (or even on smartphones), or directly exploiting the spatial information obtained by sensors mounted on the AR visor. Mixed reality (MR) refers more generally to the merging of real and virtual worlds to produce new environments and visualisations where physical and digital objects co-exist and interact in real time [22].

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 127–139, 2018.
https://doi.org/10.1007/978-3-319-94580-4_10

AR and MR technologies can have an important impact from an application point view, allowing for reshaping the environments where people work and live, rethinking the way in which they interact and collaborate [4,23]. In that perspective, it is interesting to consider how these technologies can be put in synergy with pervasive computing/ubicomp technologies [21,24], and then Internet of Things (IoT) and Web of Things (WoT) as main ingredients of modern smart environments and spaces [3,12,16].

Agents can play a key role in AR/MR for modelling and implementing holograms that need to feature an autonomous behaviour, eventually interacting with other holograms and with humans, as well as with the physical world where they are immersed. More generally, one can envision future agent-based mixed reality systems where agents can dynamically create and control virtual objects and holograms, along with the control of physical things, so as to devise new kind of smart environments [17]. Besides individual holograms, multi-agent systems and agent organisations can be fruitfully exploited to model complex and possibly open, large-scale systems of holograms, featuring some degree of coordinated, cooperative, social behaviour.

In order to flexibly exploit existing agent/multi-agent models and technologies to this purpose, we argue the need of proper conceptual and technological frameworks. In this paper, we describe a novel framework called *Augmented Worlds* (AW) to develop agent-based pervasive mixed reality systems. The framework provides a set of features that – as a whole – aims at representing a significant contribution with respect to the state-of-the-art, in particular:

- *Shared/multi-user worlds* – the framework allows for multiple human users being immersed in the same AW, sharing and possibly interacting with the same augmented (virtual) entities.
- *Bidirectional augmentation* – the framework is based on a notion of "augmentation" which is wider than the pure AR/MR one, by integrating also the pervasive computing perspective where augmentation is about enriching the physical environment with computational capabilities.
- *Open systems* – the framework makes it possible to develop agent-based AR/MR systems where agents dynamically join (and quite from) an AW and where the structure of the AW – in terms of set of virtual entities/holograms – can be dynamically changed by agents at runtime.
- *Heterogeneous agent (programming) technology* – the design of the framework allows for different, heterogeneous agent technologies to be used for programming the agents working inside the augmented world. Examples are ASTRA [8], Jason [6], JaCaMo [5].
- *Cognitive agency* – in spite of the support for heterogeneous agent technologies, the framework has been conceived to be particularly effective for cognitive agents, in particular to investigate the engineering of AR/MR systems based on BDI (Belief-Desire-Intention) multi-agents systems.
- *Enabling AR technologies* – the design of the framework allows for fully reusing and exploiting existing AR enabling technologies – both software (e.g., Unity 3D, Vuforia) and hardware (e.g., Hololens, Meta2) – yet keeping a strong separation between such enabling level and the agent level.

– *Generality* – the framework aims at being sufficiently general to support the development of effective AW for different application domains—including both indoor and outdoor scenarios.

The remainder of the paper is organised as follows: Sect. 2 provides an overview of the conceptual model of AWs and of a concrete framework implementing the idea; Sect. 3 shows an example of AWs development and Sect. 4 describes the case studies adopted so far to start to discuss the idea. Finally Sect. 5 provides an overview of related works and some concluding remarks.

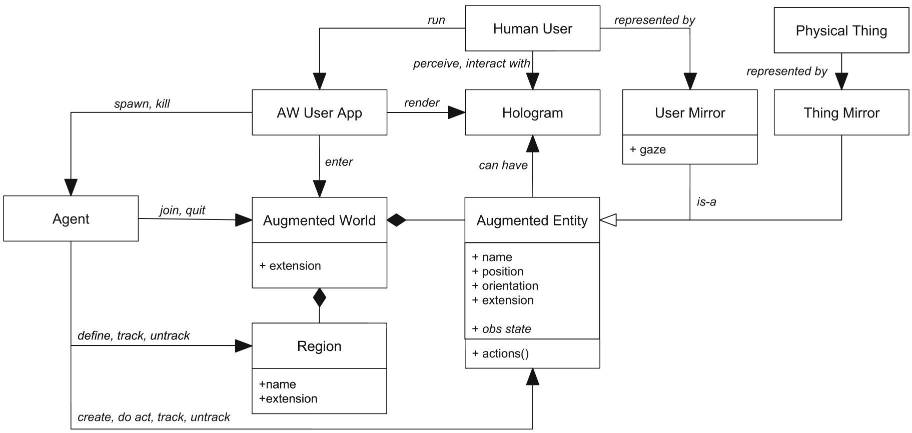


Fig. 1. Conceptual model of an augmented world.

2 The Augmented World Model and Framework

The conception of AW has been strongly influenced by the research in environments as first-class abstraction of MAS [25] and our experience with *mirror worlds* [17]—which are strongly related to AW. Accordingly, we model an agent-based pervasive mixed-reality system in terms of a dynamic set of software agents – possibly running on different hosts, including human wearable/mobile devices – working inside an application environment, possibly distributed as well. The application environment is composed by a dynamic set of virtual objects referred as *augmented entities*. Augmented entities are the basic bricks to shape the mixed reality environment, embedding some observable state and computational behaviour, and being situated in a specific position of the physical space. Agents can dynamically create, control, observe augmented entities. Some of these entities can have an AR representation, that we refer as *hologram*, that can be (shared and) perceived by human users immersed in that physical environment (by means of suitable AR devices).

The conceptual model has been strongly inspired by the A&A meta-model [15]. In fact, augmented entities can be conceptually modelled as *artifacts*

Table 1. Augmented worlds main primitives list.

Primitive actions	Description
joinAW(name, location): awID	To join an existing augmented world, getting an id of the session
quitAW(awID)	To quit from working in an augmented word
createAE(awID, name, template, args, config): aeID	To create a new augmented entity in a specified augmented world, specifying its name, template, parameters (that depend on the specific template), and initial configuration (including position, orientation, ...)
disposeAE(aeID)	To dispose an existing augmented entity
trackAE(aeID)	To start tracking an existing augmented entity
stopTrackingAE(aeID)	To stop tracking an existing augmented entity
moveAE(aeID, pos, orientation)	To change the position and orientation of an augmented entity, if allowed
defineRegion(awID, name, region)	To define a named region, specifying name and extension
trackRegion(awID, name)	To start tracking a region
stopTrackingRegion(awID, name)	To stop tracking a region

in A&A, used as first-class abstraction to design and develop the application environment where agents are situated. Beyond this, the AW conceptual model and framework introduce further concepts and features, detailed in the following, specifically tailored for pervasive mixed reality systems, such as – for instance – the spatial coupling with the physical reality and the explicit modelling of human users—which are not part of the A&A meta-model.

2.1 Conceptual Model

The main concepts of AW are formalised in the UML diagram shown in Fig. 1. An Augmented World is a virtual/computational world mapped on to some specific physical region (region attribute).

Augmented Entities. An Augmented World is composed by a set of Augmented Entity, representing the concrete (virtual) computational objects layered on top of the physical world. An Augmented Entity has a specific position, orientation and extension in the physical world (position, orientation, extension attributes), which are defined with respect to the region and system of reference defined by the Augmented World. Augmented entities are the basic bricks of the agent application environment. An Augmented Entity exposes an observable state (obs state attribute) – in terms of a set of observable properties, that can change over time – and an action interface (actions() operations). Augmented entities are can be dynamically created and disposed by the agents, as well as moved in different point of the augmented world. **Agents**. In order to work inside an Augmented World, an agent has to *join* it, setting up a working session. Once joined an Augmented World, an agent can act upon an Augmented Entity, through the actions that the Augmented Entity makes it available, and *track* it, to perceive its observable state and events. Besides tracking specific augmented entities, agents can track *regions* inside Augmented World, as portions of the physical space on which the Augmented World is mapped. As soon as an augmented entity enters or leaves a region tracked by an agent, the agent perceives a corresponding event, carrying on information about the Augmented Entity source of the event.

The set of actions that an agent can do inside an AW can be grouped in two main categories. The first one is fixed set of primitives providing predefined functionalities (see Table 1): to join and quit an AW, to create and manage augmented entities, and to define and track regions. Besides this fixed set, by joining an AW the set of available actions is given by the collection of all action interfaces provided by the augmented entities actually created in that AW. So given an augmented entity aeID providing some action (operation) op, an agent has an action: doAct(aeID,op,args) which triggers the execution of the operation on the augmented entity. As in the case of environment programming abstractions [18], actions can be long-term processes whose execution occurs inside the augmented entity, eventually completing with a success or a failure. These action events are perceived by the agent triggering the action asynchronously.

On the perception side, in an AW an agent can track either specific augmented entities (`trackAE` primitive) or regions (`trackRegion`). By start tracking an Augmented Entity, an agent continuously receives percepts about its observable state and events generated. By tracking a region, an agent perceive events related to augmented entities entering or exiting, dynamically discovering their identifiers.

In cognitive agents based on the BDI model, beliefs about the state of the Augmented World and Augmented Entity are automatically created/updated/removed depending on what an agent is tracking.

Holograms. The bridge with AR is given by the Hologram concept (see Fig. 1). An augmented entity can have an *hologram* associated: the hologram is the AR representation of the Augmented Entity, to be perceived by the human users. This representation can be as simple as a 2D text, or a complex 3D structure. The key point is that – in an AR perspective – this representation is anchored (*registered*, in AR terms) to the physical world. The hologram model (state) depends on the state of the corresponding Augmented Entity and is kept updated every time the Augmented Entity state is changed. The hologram view – i.e., the actual rendering in the AR view – depends on the hologram state and the specific capabilities of the AR devices used by the user. Besides the representation, a hologram is what enables the interaction with the human users, in terms of gesture and gaze.

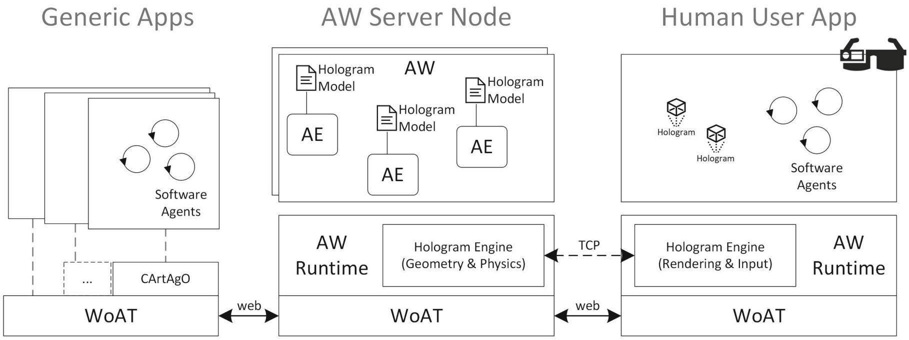


Fig. 2. Logical architecture of the AW framework.

Users. A human user starts a session inside an Augmented World by means of an AW user applications (AWUserApp). A AWUserApp spawn the agents that concretely can act inside the AW. From an UI point of view, the AWUserApp is responsible to create an AR view, given the current position and orientation of the user. Besides, this is the part in charge of detecting user inputs (e.g. gazing, gestures, voice commands). Such commands could be targeted towards specific holograms (e.g., grabbing a hologram, gazing a hologram). A hologram is then a source of observable events that can be perceived by agents, tracking the corresponding Augmented Entity. Multiple human users can be immersed in the same Augmented World, possibly using different AWUserApp.

Physical World Coupling. Some Augmented Entity can be used to represent physical objects (that are part of the physical environment) in the Augmented World, as a kind of mirror, so that the observable state of the Augmented Entity represents a model of the (physical) state of the physical thing. In Fig. 1 this is represented by the Mirror Thing concept, which is a specialisation of Augmented Entity. The coupling between the two levels – so that e.g. the observable state of the Augmented Entity is updated as soon as the physical one changes – is realised by proper *augmented entity drivers* (similar to device drivers), typically exploiting proper sensors and embedded technology. Then, by tracking an augmented entity coupled with some physical thing, an agent can perceive the physical state of the thing—as abstracted by the Augmented Entity. Among the physical things that can be coupled to an Augmented Entity, a main case is given by the physical body of a human user, so that agents can track the position of human users. This is represented by the User Mirror concept, which is specialisation Augmented Entity like in the case of Mirror Thing. The coupling between Augmented Entity and physical things can work also in the opposite direction, that is actions on the Augmented Entity can be useful to have an effect on physical things, by exploiting proper actuators. So in this case an agent can have an effect upon a physical thing of the environment by acting on the corresponding Augmented Entity.

2.2 A Prototype Framework and Infrastructure

On the basis of the AW conceptual model, we designed a first prototype framework and infrastructure for developing and running agent-based AR/MR systems. Figure 2 shows a representation of the logical architecture of the infrastructure.

From a structural point of view, we can logically recognise three main components:

- The **AW-Runtime**, which provides the infrastructure to execute AW instances, managing the augmented entities execution and providing a common interface for agents to interact and observe them. The runtime is designed to run on a server node (possibly distributed in cloud architecture), hosting the execution of one or multiple instances of AWs.
- the **Hologram Engine**, which supports holograms visualization and keeps updated their geometries in all human users devices, considering physics and managing inputs/actions of users over holograms. The main task of this component is to keep holograms (their geometries and properties) updated and consistent in each user application that wants to interact with the AW. It wraps and exploits enabling AR technologies. It runs both on the AW node and on each user device, directly communicating by means of e.g. TCP channels. From the AW point of view, the Holograms Engine provides the support to design the geometry of each hologram and keeps updated its representation according to entities properties. From the user perspective it provides all features needed to display the hologram according to the Augmented/Mixed Reality view, to manage physics (e.g. collisions between two holograms or perspective issues between holograms and real objects), and to manage user inputs and interactions with holograms—e.g. allowing to exploit gaze as a form of interaction or detecting hands/fingers gestures to manipulate holograms.
- the **WoAT** layer, which enables full interoperability at the application level. The architecture has been conceived by considering the requirements defined in Sect. 1, in particular: *(i)* full interoperability and openness at the application level, to be open to any technology for implementing agents working inside augmented worlds, yet enforcing an agent-oriented level of abstraction at the design level; *(ii)* distribution and scalability in the model adopted for defining and structuring the set of augmented entities. *(iii)* clean integration with the Internet of Things world. To this purpose, we adopted the Web of Things model proposed in the context of IoT [10], so that augmented worlds and entities (from an agent point of view) are network reachable resources providing a Web REST API to interact with them. That is, the interaction with augmented entities is mediated by a RESTful interface based on self-descriptive messages, HTTP operations (GET, POST, PUT, DELETE, HEAD) and event-oriented mechanisms like, e.g., web sockets. Each entity can be reached by means of a proper URL – related to the AW root URL – and (1) GET operations can be used to retrieve observable properties while

(2) POST operations can be used to send actions to be executed on the augmented entity. This interaction layer is referred as *Web of Augmented Things* (WoAT) [9]—the bottom layer in Fig. 2.

Current prototype is available[1] as open-source technology. The AW-Runtime is written in Java, using Vert.x[2] library to implement (part of) the WoAT layer. The Hologram Manager is based on Unity 3D equipped with the Vuforia plugin to manage AR aspects and with built-in scripts developed in C# to provides specific features for AW. A Java-based API is provided for developing the template of the augmented entity, allowing to create many instances of the same template. A taste of the API is provided in next section.

On the agent side instead, the AW any agent platform can be adopted to develop and run agents—this to promote interoperability and the vision of Web of (Augmented) Things. The Environment Interface Standard (EIS) [2] can be used to create the interface between existing agent programming platforms (e.g. GOAL, ASTRA) and the AW platform. In our case, we adopted the JaCaMo [5] platform to develop and run examples and case studies, described in next sections. Accordingly, the cognitive BDI-based agents are implemented in Jason and a set of artifacts – based on CArtAgO – has been exploited to ease the integration with the AW platform.

3 Developing AW with Cognitive Agents: A Taste

The aim of this section is to give a taste AW design and development, by considering a simplified version of a real-world case study, an *augmented museum* (discussed in next section). Figure 3 shows an abstract representation of the AW setting, along with the main architectural elements (agents and augmented entities) involved in the example, while Fig. 6 in next page shows a snapshot of the real-world setting. The AW is composed by an ancient Roman boat autonomously navigating over a virtual sea place among a set of real ancient amphorae, arranged in circle. If/when the boat hits a physical amphora, it goes back and changes direction. If/when the boat enters into a specific region ("red-zone") of the virtual sea, then a physical small light house switches on the light, which are switched off as soon as the boat lefts the zone. Multiple human visitors (users) can enter the AW, sharing the experience and interact with the boat. Visitors can generate some wind, affecting the trajectory of the boat. The coupling between the AW and physical world – i.e., the "registration" in AR terms – is based on Vuforia markers placed in environment, at the center of the stage.

Even if quite simple and classic, this example involves some of the main key points about AW: holograms featuring an autonomous behaviour interacting with the physical world, bi-directional augmentation, multi-user system.

1 https://bitbucket.org/account/user/awuniboteam/projects/AW.
2 http://vertx.io.

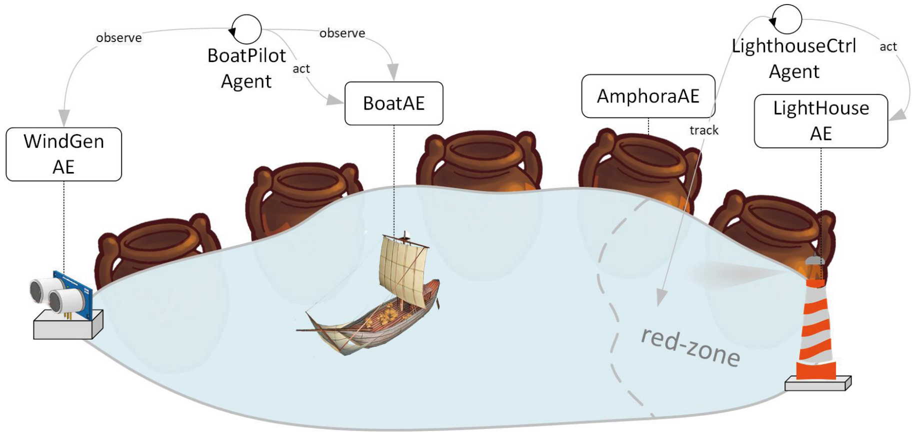


Fig. 3. Augmented museum example: AW design and main elements.

```
@HOLOGRAM("Boat")
class Boat extends AE {

  @PROPERTY double speedVal;
  @PROPERTY Vector2D speedVersor;
  @PROPERTY double windForce;

  @ACTION
  void setSpeed(Vector2D versor, double val) {
      customProperty("speedVal", val);
      customProperty("speedVersor", versor);
  }

  @ACTION
  void setWindForce(double windForce) {
    customProperty("windForce", windForce);
  }
}
```

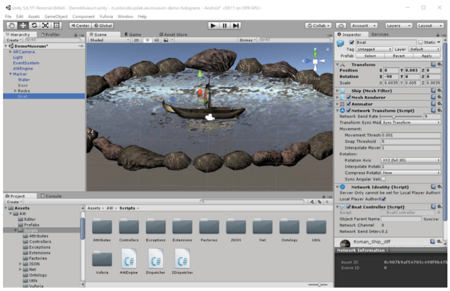


Fig. 4. (Left) The boat AE template definition. (Right) Unity screenshot with Boat prefab definition and details about AW C# scripts.

At the design level, a boat pilot agent BoatPilot encapsulates the autonomous behaviour of the boat, while the body and hologram of the boat is represented by the BoatAE augmented entity. Figure 4 shows a snippet of the source code of the BoatAE implementation based on current AW java-based API. The template of an augmented entity – defining the structure and behaviour of the instances – is mapped onto a Java class, properly annotated. In particular, `@HOLOGRAM`, `@PROPERTY` and `@ACTION` annotations can be used to identify, respectively, the Unity 3D prefab defining the geometry of the hologram, the observable properties and the actions of the entity. In the example, the BoatAE augmented entities have an hologram called Boat—shown in Fig. 4, right side.

Figure 5 shows a snippet of the code of the BoatPilot agent, implemented in Jason. The goal of the agent is to continuously move the boat according to a

```
!init.

+!init
  <- joinAW("museumAW"),
     trackAE("boat"),
     trackAE("wind"),
     !navigate(30).

+!navigate(Speed)
  <- getRandomOrientation(Versor),
     setSpeed(Versor, Speed),
     .wait(5000),
     !navigate(Speed).

@manage_border_collision[atomic]
+borderReached
  <- !divertBoatRoute(180).

+windForce(V)
  <- setWindForce(V).
```

```
!init.

+!init
  <- joinAW("museumAW"),
     trackRegion("museumAW","red-zone").

+regionUpdate("red-zone", "enter", AE)
  <- turnOnLighthouse,
     registerRedZoneAccess(AE).

+regionUpdate("red-zone", "exit", AE)
  <- turnOffLighthouse,
     registerRedZoneExit(AE).
```

Fig. 5. A snippet of the Jason source code of the BoatPilot agent *(left)* and of the *LighthouseController* agent *(right).*

random trajectory and react to relevant events. The agent tracks (observes) BoatAE so as to react to a collision event. The collision event is generated when the BoatAE collides with any AmphoraAE augmented entity, which are augmented entities modeling/coupled to the real amphorae. The agent tracks also WindGenAE, to react to changes into wind force and acts on BoatAE accordingly. This WindGenAE augmented entity is coupled to a physical Thing, embedding an Arduino single-board micro-controller with a proximity sensor, to detect the distance of user hands, simulating the wind. In a Web-of-Thing style, this thing can be accessed using a REST web API—used in the WindGenAE implementation. To that purpose, the embedded system uses a Bluetooth connection to a gateway, based on Raspberry PI 3, functioning as a bridge to the Internet.

The MAS includes also the LightHouseControllerAgent (Fig. 5, right), which is tracking the region corresponding to the red-zone and switches on/off the light

Fig. 6. The snapshots from the Augmented Museum—a visitor watching the boat moving in the virtual sea placed within a set of ancient amphorae arranged in circle.

by acting on the LightHouseBuildingAE augmented entity. Also this augmented entity is coupled to a physical thing, based on Arduino. This agent reacts to the entrance/exit of any augmented entity (the boat in this case) and performs switchOn/switchOff action on the augmented entity.

4 Case Studies and Discussion

The augmented museum case study is part of a cultural heritage project, in cooperation with an Italian Museum in the context of maritime archaeology[3]. Generally speaking, the objective of the project is to investigate the design of augmented worlds layered upon two existing physical environments – a room in a museum and an harbour – so as to enrich the experience of the visitors. In the project involves two case studies: the augmented museum[4] – useful for us to start applying concretely the idea in the case of *indoor* scenarios – and the *augmented harbour*), which involves an *outdoor* scenario.

The augmented harbour – whose development is ongoing – accounts for designing an augmented world enriching a physical harbour with a group of virtual old fishing boats moving around in the sea and periodically entering the harbour. Visitors are meant to enjoy the exhibition from some fixed sightseeing points placed along the harbour side. Each sightseeing point includes a tablet with the camera pointing the (augmented) sea, allowing for some rotation to look around. In this scenario, the physical position of the boats as augmented entities with a hologram is given by a GPS device. As in the other case study, each boat is controlled by a pilot agent with the goal to mange boat route avoiding to bump other virtual boats entering and navigating in the harbour and avoiding to collide with harbour borders. To detect such collisions, the harbour borders are explicitly represented inside the AW by means of mirror augmented entities. This case study is useful to stress more the multi-agent aspect—possibly involving several boats moving around interacting among them and with the physical world.

Indeed, these are quite simple and “classic” examples of mixed/augmented reality systems. Currently we are looking for case studies to better stress aspects such as: dynamism—with agent dynamically manipulating (creating, disposing) virtual objects in the mixed environment; complexity of the cognitive agency—involving more complex autonomous behaviours for the agents; and user interaction—exploiting in particular the possibility for agents inside the AW to track user position and gaze.

5 Related Works and Conclusions

In literature, a comprehensive description about the role of agents and MAS in Mixed and Augmented Reality contexts has been already given with the

[3] Museo della Regina, Cattolica (RN), Italy.
[4] https://goo.gl/avBSgH.

concept of MiRa (Mixed Reality Agents) [11] and AuRAs (Augmented Reality Agents) [7]. The focus of the investigation of MiRA and AuRA is about agents having a representation in an AR/MR environment making them perceivable by human users and enabling interaction with them, as well as with other agents. Our work is aimed instead at defining a suitable conceptual model and technological framework for developing and executing agent-based AR/MR systems, featuring various degrees of distribution, autonomy, interaction with the physical world and flexibly exploiting (cognitive) agents as the key building block to design the autonomous parts.

Another main related work is given by Mirror Worlds [17,19], which can be considered the root of the AW idea. The AW conceptual model (described in Sect. 2) is an extension and evolution of the MW one, towards its concrete adoption in the design and engineering of real-world agent-based AR/MR systems. Besides, the design of the AW infrastructure is explicitly tailored to maximise interoperability with the Internet-of-Things as bridge to the physical world.

Finally, the literature on Intelligent Virtual Environments (IVEs) [13] is a further main related work of ours, in particular those contributions extending the basic IVEs towards the integration with the physical environment (e.g., [20]). Compared to these approaches, Augmented and Mixed Reality are treated as first-class aspects in the AW model and the platform, which is not the case of IVEs—more related to Virtual Reality. In this perspective, AW aims at fully enacting a form of *bi-direction augmentation* which is typically out of the scope of IVEs—integrating the Augmented Reality and Pervasive Computing, as supported by the Web of Augmented Thing idea and layer.

To sum up, the main aim of the AW framework is to provide a conceptual and practical tool for investigating and exploiting the use of cognitive agents and multi-agent systems for the development of complex augmented reality and mixed reality applications. In this paper we described *(a)* the AW conceptual model—which aims at being expressive enough to support the design of a variety of augmented worlds with different kinds of coupling between the virtual layer and the physical world; and *(b)* a first concrete platform, to explore in practice the development of AW for specific application domains. Future work will be devoted to refine and improve the architecture and prototype implementation of the framework, in particular by tackling more challenging case studies to get proper feedbacks to that purpose.

References

1. Azuma, R.T.: A survey of augmented reality. Presence: Teleoper. Virtual Environ. **6**(4), 355–385 (1997)
2. Behrens, T.M., Hindriks, K.V., Dix, J.: Towards an environment interface standard for agent platforms. Ann. Math. Artif. Intell. **61**(4), 261–295 (2011)
3. Billinghurst, M., Clark, A., Lee, G.: A survey of augmented reality. Found. Trends Hum.-Comput. Interact. **8**(2–3), 73–272 (2015)
4. Billinghurst, M., Kato, H.: Collaborative augmented reality. Commun. ACM **45**(7), 64–70 (2002)

5. Boissier, O., Bordini, R.H., Hübner, J.F., Ricci, A., Santi, A.: Multi-agent oriented programming with JaCaMo. Sci. Comput. Program. **78**(6), 747–761 (2013)
6. Bordini, R.H., Hübner, J.F., Wooldrige, M.: Programming Multi-agent Systems in AgentSpeak using Jason. Wiley Series in Agent Technology. Wiley, Hoboken (2007)
7. Campbell, A.G., Stafford, J.W., Holz, T., O'hare, G.M.: Why, when and how to use augmented reality agents (AuRAs). Virtual Real. **18**(2), 139–159 (2014)
8. Collier, R.W., Russell, S., Lillis, D.: Reflecting on agent programming with AgentSpeak(L). In: Chen, Q., Torroni, P., Villata, S., Hsu, J., Omicini, A. (eds.) PRIMA 2015. LNCS (LNAI), vol. 9387, pp. 351–366. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25524-8_22
9. Croatti, A., Ricci, A.: Towards the web of augmented things. In: 2017 IEEE International Conference on Software Architecture Workshops (ICSAW), pp. 80–87, April 2017
10. Guinard, D.D., Trifa, V.: Building the Web of Things. Manning, New York (2016)
11. Holz, T., Campbell, A.G., O'Hare, G.M.P., Stafford, J.W., Martin, A., Dragone, M.: Mira-mixed reality agents. Int. J. Hum.-Comput. Stud. **69**(4), 251–268 (2011)
12. Lasi, H., Fettke, P., Kemper, H.G., Feld, T., Hoffmann, M.: Industry 4.0. business & information. Syst. Eng. **6**(4), 239–242 (2014)
13. Luck, M., Aylett, R.: Applying artificial intelligence to virtualreality: intelligent virtual environments. Appl. Artif. Intell. **14**(1), 3–32 (2000)
14. Milgram, P., Kishino, F.: A taxonomy of mixed reality visual displays. IEICE Trans. Inf. Syst. **E77-D**(12), 1321–1329 (1994)
15. Omicini, A., Ricci, A., Viroli, M.: Artifacts in the A&A meta-model for multi-agent systems. Auton. Agents Multi-agent Syst. **17**(3), 432–456 (2008)
16. Panetta, K.: Exploring augmented reality for business and consumers. Gartner report (2017)
17. Ricci, A., Piunti, M., Tummolini, L., Castelfranchi, C.: The mirror world: preparing for mixed-reality living. IEEE Pervasive Comput. **14**(2), 60–63 (2015)
18. Ricci, A., Piunti, M., Viroli, M.: Environment programming in multi-agent systems: an artifact-based perspective. Auton. Agent. Multi-agent Syst. **23**(2), 158–192 (2011)
19. Ricci, A., Tummolini, L., Piunti, M., Boissier, O., Castelfranchi, C.: Mirror worlds as agent societies situated in mixed reality environments. In: Ghose, A., Oren, N., Telang, P., Thangarajah, J. (eds.) COIN 2014. LNCS (LNAI), vol. 9372, pp. 197–212. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25420-3_13
20. Rincon, J.A., Poza-Lujan, J.L., Julian, V., Posadas-Yague, J.L., Carrascosa, C.: Extending MAM5 meta-model and JaCalIVE framework to integrate smart devices from real environments. PLoS ONE **11**(2), 1–27 (2016)
21. Satyanarayanan, M.: Pervasive computing: vision and challenges. IEEE Pers. Commun. **8**, 10–17 (2001)
22. Schmalstieg, D., Höllerer, T.: Augmented Reality: Principles and Practice. Addison-Wesley, Boston (2015)
23. Starner, T.: Project glass: an extension of the self. IEEE Pervasive Comput. **12**(2), 14–16 (2013)
24. Weiser, M.: The computer for the 21st century. SIGMOBILE Mob. Comput. Commun. Rev. **3**, 3–11 (1999)
25. Weyns, D., Omicini, A., Odell, J.: Environment as a first class abstraction in multiagent systems. Auton. Agent. Multi-agent Syst. **14**(1), 5–30 (2007)

Classification of Spatio-Temporal Trajectories Based on Support Vector Machines

Jesus Cuenca-Jara[1], Fernando Terroso-Saenz[2], Ramon Sanchez-Iborra[1(✉)], and Antonio F. Skarmeta-Gomez[1]

[1] Department of Information and Communications Engineering, University of Murcia, Murcia, Spain
{jesus.cuenca1,ramonsanchez,skarmeta}@um.es

[2] Department of Computer Science, Catholic University of Murcia, Murcia, Spain
fterroso@ucam.edu

Abstract. Within the mobility mining discipline, several solutions for the classification of spatio-temporal trajectories have been proposed. However, they usually do not fully consider the particularities of trajectories from human-generated data like online social networks. For that reason, this work introduces a novel classifier based on Support Vector Machines (SVM), which fits the low resolution of this type of geographic data. This solution is applied in a use case for the detection of tourist mobility exhibiting quite promising results.

Keywords: Trajectory classification · Support Vector Machine Tourist mobility

1 Introduction

Last decade has witnessed the dawn of personal mobile contrivances as the center of our digital life at the same time that manufacturers endlessly empowered such devices with new and more advanced sensing features. One clear example of this enrichment is the fact that now mobile devices are commonly equipped with outdoor positioning technologies such as GPS and other sensors that might provide indoor location like RFID or Bluetooth.

Citizens generate an unprecedented amount of spatio-temporal data through ubiquitous use of location-aware personal devices [2]. Furthermore, all this data can be easily hosted and shared in different community-based platforms. In that sense, Online Social Networks (OSNs), like Twitter[1], Facebook[2] or Flickr[3], have been some of the ecosystems that have contributed most to the proliferation of such user-generated content [1].

[1] https://twitter.com/.
[2] https://www.facebook.com/.
[3] https://www.flickr.com/.

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 140–151, 2018.
https://doi.org/10.1007/978-3-319-94580-4_11

The wealth of available OSN-based mobility data is one of the key factors that has allowed the fast development of the mobility mining field. In general terms, we can distinguish three types of data analysis within the aforementioned discipline [12], (i) clustering (grouping geographical or spatio-temporal data sharing similar features), (ii) pattern discovery (detecting regularities of movement behaviours) and (iii) classification (labelling geographical or spatio-temporal data with respect a predefined set of classes, e.g. the means of transport associated to a trajectory).

Whilst a plethora of data-mining algorithms has been successfully applied to OSN-based data in the context of the first two types of analysis [3,6,7], some classification algorithms have been proposed to particularly deal with spatio-temporal trajectories [8,9,11]. These algorithms relied on high-resolution mobility data where the moving object frequently reports its current location. This type of data is usually generated by certain GPS feeds like navigators. However, OSN data is often characterized to provide more coarse-grained trajectories as users tend to sparsely report their location data.

All of this has led us to present a novel approach to classify spatio-temporal trajectories which considers its coarse-grained nature. In particular, the present work states a methodology that combines density clustering and Support Vector Machines (SVMs) [13] to compose the final classifier.

As a particular operational scenario, we apply the aforementioned methodology to the detection of tourist and resident mobility flows in an urban environment. This way, the resulting system is able to classify OSN users as a tourists or residents of a city by processing the geo-tagged content that they submit to a certain OSN platform.

The remainder of the paper is structured as follows. Section 2 is devoted to describing in detail the logic structure and the processing stages of the proposed system. Then, Sect. 3 discusses the main results of the performed experiments. Next, an overview about mobility data classification and tourist mobility analysis is put forward in Sect. 4. Finally, the main conclusions and the future work are summed up in Sect. 5.

2 Related Work

When it comes to classify spatio-temporal trajectories, existing literature usually follows a three-step procedure [16], (i) a segmentation step where the trajectory is split into different parts sharing certain similarities, (ii) extract certain features for each segment and (iii) generate a model able to classify each segment given a set of pre-defined classes.

Given this general framework, most previous works focused on studying the suitability of different models for the third step. In that sense, Hidden Markov Model (HMM) or Decision Trees (DT) have been common choices [2]. For example, the work in [19] used a HMM to detect the status of a taxi in terms of *occupied*, *non-occupied*, and *parked* on the basis of its GPS trajectories. Regarding DT, the proposal in [17] stated a DT solution to a similar transportation-mode detection.

Other works have gone beyond the aforementioned procedure and provided more robust and sophisticated solutions for trajectory classification. This is the case of the *TraClass* framework [9], which relies on the hypothesis that the best discriminative features might appear as small regions covered by a trajectory. Hence, it makes use of a variant of *TraClus*, a well-known trajectory clustering algorithm [10], and a hierarchical region-based clustering to translate the spatio-temporal trajectories to feed a HMM providing the final outcome. Besides, the *TRAOD* framework [8] provides a trajectory classification method to detect abnormal trajectories so-called outliers. Finally, the work in [11] proposed a novel trajectory representation based on geographic *motifs*. These motifs gave raise to a new feature space that was used to build a hierarchical-based classifier.

One common feature of all the aforementioned works is the requirement to properly segment incoming trajectories to further analyse them. In that sense, this segmentation process assumes that the trajectories temporal granularity is quite fine, in the order of seconds or minutes. Consequently, such works fit well in scenarios where GPS feeds automatically gather the current location of the target moving objects. However, this granularity does not apply for OSN data, where content is supplied from time to time, as it encompasses a wide range of activities, ranging from the leisure activities of locating summer holiday photographs to those focused on surveying in the aftermath of an earthquake.

Unlike such proposals, the present work provides a trajectory classification solution that is compatible with such data sparsity problem. In that sense, the representation of individual trajectories as sequences of clusters used in this work has been already proposed [18]. Nonetheless, in such work, clusters are generated with a hierarchical clustering, whereas in our case we apply a density-based clustering.

3 Methodology

Figure 1 outlines the main steps to come up with the proposed classifier. The first step is to collect the dataset that will constitute the training set to generate and tune the final classifier. After that, the individual trajectories are extracted. The third step is to identify meaningful geographic areas through density-based clustering. Next, we map the individual trajectories on the basis of such areas. Lastly, these routes are the basis to create the final SVM.

Here we put forward the details of each step of the proposed methodology.

3.1 The Classification Task

The objective of this work is to obtain an automatic mechanism in such a way that every given trajectory is classified as one among a set of pre-defined labels or classes. Therefore, the first step is to define what is the set of predefined and meaningful classes. Let it be $\Omega = \{CL_1, CL_2, \ldots, CL_p\}$.

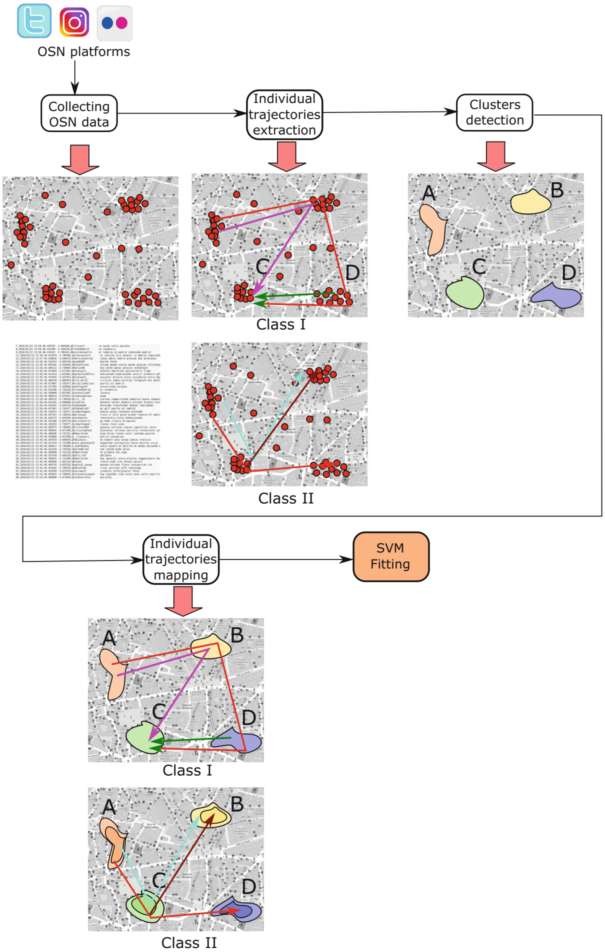


Fig. 1. Approach overview.

3.2 OSN Data Collection and Filtering

The first step of the methodology is to gather OSN geo-tagged data from different platforms. In that sense, many of these platforms already provide open Application Programming Interfaces (APIs) that can be used in order to gather their publicly visible documents. Depending on the platform under consideration, a OSN *document* will take different formats. For example, in case of OSN platforms, it will be geo-tagged *tweets* in the case of Twitter or photographs in Flickr.

Due to this variety, the present work proposes an uniform view of the gathered OSN data. Hence, a OSN document $d \in \mathcal{D}$ is characterized by a user u who actually generated the document, the spatial location l at which d was posted as a pair $\{x, y\}$ of coordinates and the timestamp of the submission t. Thus, a document is defined as a tuple $d = \langle u, l, t \rangle$

Once the extraction of OSN geo-tagged data has been completed, it is necessary to clean the collected dataset, so that it eventually contains effective human mobility information. For this reason, for each pair of consecutive documents posted by the same user u, (d_i^u, d_{i+1}^u) if their spatial and temporal difference is below a threshold $\langle \Delta_l, \Delta_t \rangle$ only the first one d_i^u is filtered in whereas the second one d_{i+1}^u is discarded. The rationale of this filtering is to only keep documents that actually represent meaningful spatial movements of users. As a result, a dataset $\mathcal{D}_f \subseteq \mathcal{D}$ of filtered OSN documents is generated.

3.3 Trajectories Extraction and Labelling

The next step focuses on the extraction of spatio-temporal routes on the basis of the filtered documents. To do so, we *slice* the documents in $\mathcal{DC}_f$ belonging to the same user u in sequences of consecutive documents posted during the same day. Accordingly, we can define a trajectory tr_{day}^u as a sequence of consecutively visited locations by a user u during the same day, $tr_{day}^u = \{v_{l_0,t_0}^u \rightarrow v_{l_1,t_1}^u \rightarrow .. \rightarrow v_{l_n,t_n}^u\}$, $n \geq 2$, where $v_{l_i,t_i}^u \in \mathcal{L} \times \mathcal{T}$ is the location and timestamp of the i-th clean document posted by the user u during day, so that $t_i < t_{i+1} \forall i \in [0, n]$.

Then, each trajectory is labelled, manually or semi-automatically, with one of the possible classes that our classifier will be able to detect, represented by the set $\Omega = \{CL_1, CL_2, \ldots, CL_p\}$. As a result, a set $\mathcal{TR}$ of labelled trajectories is made up. Each element of this set takes the form of $\langle tr_{day}^u, CL \rangle$ where tr_d^u is a user trajectory and CL its associated class. At this point, $\mathcal{TR}$ constitutes the training dataset that will be used during the following steps to generate the final SVM whose main goal will be, given certain trajectory of certain user, assigning the corresponding label $CL_i \in \Omega$.

3.4 Clustering Process

Once the training dataset has been composed, we generate the clusters representing meaningful spatial regions in the area of interest by means of a density-based clustering algorithm. In short, density-based clustering is based on the concept of *local neighbourhood* $\mathcal{N}$ of a point p, that is, the number of points that are within a certain straight distance *Eps* to p,

$$\mathcal{N}(p) = \{q \in \mathcal{S} \mid dist(p, q) \leq Eps\}$$

where $\mathcal{S}$ is the set of available points. If $|\mathcal{N}(\text{p})|$ is over a certain threshold $MinPoints$, then $\mathcal{N}(\text{p})$ is considered a cluster. Furthermore, $\mathcal{N}(\text{p})$ is *density-joinable* to $\mathcal{N}(\text{q})$ (p$\neq$q) if $\mathcal{N}(\text{p}) \cap \mathcal{N}(q) \neq \emptyset$

The density clustering algorithm applied for this task is the *landmark discovery algorithm* (LDA) described in [14]. Briefly, LDA is a modified version of the well-known DBSCAN [4] algorithm that provides an extra level of indexation over the discovered clusters called *landmarks*. A *landmark* is defined as a group of centroids where the neighbourhood of each centroid is density-joinable with the neighbourhood of at least other centroid in the same landmark as Fig. 2 illustrates.

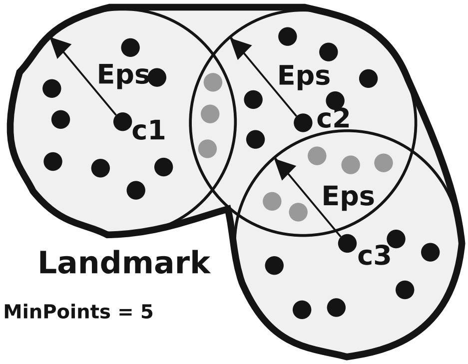


Fig. 2. Example of a landmark returned by the LDA.

Finally, the clustering task using the LDA is performed in the set of individual labelled locations obtained "decomposing" the pairs $\langle tr^u_{day}, CL_i\rangle$. After this clustering process, a set of representative locations is obtained.

3.5 Trajectories Mapping

According to Fig. 1, after generating the clusters (landmarks), we map the individual trajectories in $\mathcal{TR}$ (see Sect. 3.3) to the space defined by the clusters. To do so, for each trajectory $tr^u_{day} \in \mathcal{TR}$, we translate each location $v^u_{l,t}$ to its closer cluster $c \in \mathcal{C}$. As a result, each trajectory tr^u_{day} based on visited locations, gives rise to a new trajectory trc^u_{day} defined as a sequence of visited clusters $trc^u_{day} = \{id_{c_0} \rightarrow id_{c_1} \rightarrow \ldots \rightarrow id_{c_n}\}$ where id_{c_i} is the identifier of the i-th visited landmark by the user u during a day.

On the basis of this cluster-based approach, a new dataset $\mathcal{TRC}$ of labelled trajectories is generated. Each element of this dataset takes the form of $\langle trc^u_{day}, CL\rangle$ where trc^u_{day} is a individual cluster-based trajectory and $CL \in \Omega$ the associated class or the original trajectory tr^u_{day}.

3.6 Generation of the SVM Model

The dataset $\mathcal{TRC}$ is used to fit the SVM algorithm (with RBF kernel) that will perform the classification task. SVM is a classification algorithm that works with hyperplanes to separate data into different classes [13]. Each hyperplane divides the data in two different classes and tries to leave the maximum margin from both.

4 Evaluation

We have evaluated the feasibility of our methodology in a real use case scenario. The goal of this scenario is to classify the urban trajectories extracted from a OSN platform to know if they are related to tourist (TR) or local citizens (LC). This classification may be used to provide extra-valued services to both locals and visitors in real time.

4.1 Dataset

The use case covers the large metropolis of New York (NY) in the United States. We extracted from Flickr, a well-known social platform for photo-sharing, a OSN dataset from the city. More in detail, we have used documents contained in the Yahoo Flickr Creative Commons 100M public repository [15]. By doing this, we kept the geo-tagged documents from the repository for the city, but specifically only the ones that fit into the spatial polygon defined for the city in OpenStreetMap[4] covering a three-year period. Table 1 summarizes the details of these datasets.

Table 1. Evaluation datasets.

Feature	New York (NY)
Time period	2010/01/01 $\rightarrow$ 2012/12/31
Covered area (km^2)	1214
Geo-tagged documents/users	343952/12502

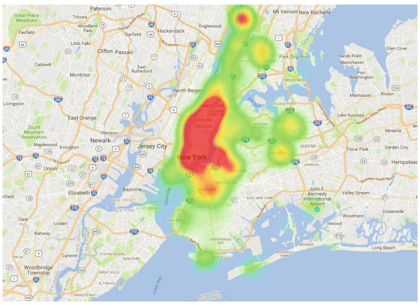


Fig. 3. Heat map of document density of the evaluated dataset.

In Fig. 3 we can observe the heat map of the dataset showing the density of OSN documents. We see the direct correlation between the density of documents and their distance with respect to the center of the city.

[4] http://www.openstreetmap.org/.

4.2 Target Classes

For the present deployment, we distinguish two types of trajectories: the trajectories of the local citizens living in the city and the trajectories of tourists. Consequently, the set of targets to classify each trajectory takes the form $\Omega = \{LC, TR\}$.

4.3 Settings

Table 2 summarizes the configuration parameters of the methodology for the present use case. Particularly, the table shows the parameters used in the OSN data filtering and the SVM model (adjustment and final configuration values).

Table 2. Parameters for the different steps of the methodology.

Step	Name	Value	Description
Data gathering	Δ_l	1500 m	Minimum distance between consecutive locations
	Δ_t	1800 s	Minimum time difference between consecutive locations
SVM adjustment	Γ_{min}	20	Minimum radial basis function-specific kernel parameter value
	C_{min}	0.2	Minimum general penalizing parameter for C-classification value
	Γ_{max}	400	Maximum radial basis function-specific kernel parameter value
	C_{max}	5	Maximum general penalizing parameter for C-classification value
	Δ_Γ	20	Radial basis function-specific kernel parameter increase
	Δ_C	0.2	General penalizing parameter for C-classification increase
SVM final config.	Γ_f	120	Final radial basis function-specific kernel parameter value
	C_f	1.4	Final general penalizing parameter for C-classification value

4.4 OSN Data Collection and Filtering Results

Before further processing the collected datasets, we applied the location filtering based on geographical and temporal distance described in Sect. 3.2. Table 3 shows the results of this filtering for the specific case of the city of New York.

Table 3. Dataset cleaning results.

Feature	NY
Discarded documents	92232 (27%)
Final documents ($\mathcal{D}_f$)	251720 (73%)

4.5 Trajectories Extraction Results

The obtained $\mathcal{D}_f$ set was used to compose the individual daily trajectories and associate them with their corresponding label in Ω. For this association, we followed a user-based approach to identify whether a user was a resident or a tourist. Then, we labelled all the trajectories of this user with the found class.

In order to detect if a user is a local resident or a tourist, we followed a time-based approach commonly accepted in the related literature [5]. Basically, it divides the study period into 30-day blocks. If the photographers took all their photos within a period of 30 days, the algorithm would label them as visitors (TR). On the other hand, if they uploaded photographs at intervals of more than 30 days, they would be categorized as residents (LC).

Table 4 shows the resulting distribution of users in these two categories.

Table 4. Distribution of users per class.

	NY
LC	2245 (35%)
TR	4353 (65%)
Total	6598

From the obtained results, we labeled each individual trajectory with the associated class of its users. Table 5 shows the distribution of trajectories regarding the target classes.

Table 5. Distribution of individual trajectories per class.

	NY
LC	8673(70%)
TR	3746 (30%)
Total	12419

The distribution of trajectories among classes is less balanced than the user's one as most of the trajectories correspond to residents. This is due to the fact that each local individual covers a longer time frame than most visitors.

4.6 Clusters Generation and Trajectory Mapping Results

The next step in the evaluation process was the generation of the different clusters. After uncovering the clusters, we performed the trajectory mapping (see Sect. 3.5). As a result, Table 6 shows the distribution of trajectories with respect to their length (number of clusters comprising their sequence). It can be seen that after the mapping procedure, most trajectories are composed of sequences of 2 or 3 clusters. This fact is consistent with the well-known feature of OSN-based trajectories regarding their inherent low-resolution.

Table 6. Distribution of individual trajectories per length.

	Length 2	Length 3	Length 4	Length 5	Length 6
LC	5931	1841	556	236	109
TR	2060	914	401	251	120

4.7 SVM Classifier Results

Once the set of routes for both users were generated, we tested the effectiveness of our method with the employment of the well-known SVM (Support Vector Machine) classifier, described in Sect. 3.6.

For this purpose, we made a cross-validation in which ten different sets of training-testing have been generated randomly. The data in each of these sets were 80% dedicated to training and the other 20% to testing. We applied different configurations for the parameters Γ and C for these sets, as shown in Table 2 in the SVM adjustment section. The best results were obtained for the Γ_f and C_f parameter values shown in Table 2.

Table 7. Confusion matrixes of SVM classification per route length.

	Length 2		Length 3		Length 4		Length 5		Length 6	
	LC	TR	LC	TR	LC	TR	LC	TR	LC	TR
LC	**0.73**	0.27	**0.80**	0.20	**0.75**	0.25	**0.64**	0.36	**0.56**	0.44
TR	0.29	**0.71**	0.23	**0.77**	0.28	**0.72**	0.33	**0.67**	0.45	**0.55**

Table 7 shows the results obtained with the SVM final configuration. This table shows, for each route-length, its relative confusion matrix. It was generated using the average values obtained of the 10 sets described previously.

Observing the confusion matrices, it can be concluded that the best results were obtained for route-lengths of 3 and 4. On the other hand, for a route-length of 2, the results were weaker due to the overlapping caused by the shortness of the routes. Finally, for lengths longer than 4, results were also less accurate because there is too little data (as seen in Table 6) even though there is less route overlapping.

5 Conclusions

The fusion of GPS-enabled personal contrivances and Online Social Networks (OSNs) has facilitated the generation of an unprecedented amount of human-generated mobility data. In that sense, most of the works in charge of classify such data when it takes the form of spatio-temporal trajectories focus on high-resolution scenarios. However, data from OSNs usually provide rather coarse-grained spatio-temporal information.

As a result, the present work provides a novel approach to label OSN-based trajectories using a Support Vector Machines (SVMs). Such an approach profits from the low-resolution of these trajectories to compose a set of features feeding the SVM. Lastly, the system have been tested in a real use case scenario comprising a large city in order to classify trajectories in terms of origin. Further work will focus on enriching the classification process with other sources of information. For example, the textual content included by users in most uploaded documents could be used to tag clusters with descriptive labels about the activity involved in their areas. Such tagging may help to differentiate clusters more clearly and, thus, improve the classification accuracy of the system.

Acknowledgments. This work has been sponsored by the Spanish Ministry of Economy and Competitiveness through the PERSEIDES project (contract TIN2017-86885-R) and by the European Union under the framework of the H2020 IoTCrawler project (contract 779852).

References

1. Bellavista, P., Montanari, R., Das, S.K.: Mobile social networking middleware: a survey. Pervasive Mob. Comput. **9**(4), 437–453 (2013)
2. Renso, C., Stefano Spaccapietra, E.Z.: Mobility Data - Modeling, Management, and Understanding. Cambridge University Press, Cambridge (2013)
3. Comito, C., Falcone, D., Talia, D.: Mining human mobility patterns from social geo-tagged data. Pervasive Mob. Comput. **33**, 91–107 (2016)
4. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. KDD **96**, 226–231 (1996)
5. Girardin, F., Calabrese, F., Fiore, F.D., Ratti, C., Blat, J.: Digital footprinting: uncovering tourists with user-generated content. IEEE Pervasive Comput. **7**(4), 36–43 (2008)
6. Han, J., Lee, H.: Adaptive landmark recommendations for travel planning: personalizing and clustering landmarks using geo-tagged social media. Pervasive Mob. Comput. **18**, 4–17 (2015)
7. Hu, Y., Gao, S., Janowicz, K., Yu, B., Li, W., Prasad, S.: Extracting and understanding urban areas of interest using geotagged photos. Comput. Environ. Urban Syst. **54**, 240–254 (2015)
8. Lee, J.G., Han, J., Li, X.: Trajectory outlier detection: a partition-and-detect framework. In: 2008 IEEE 24th International Conference on Data Engineering, pp. 140–149, April 2008

9. Lee, J.G., Han, J., Li, X., Gonzalez, H.: Traclass: trajectory classification using hierarchical region-based and trajectory-based clustering. Proc. VLDB Endow. **1**(1), 1081–1094 (2008)
10. Lee, J.G., Han, J., Whang, K.Y.: Trajectory clustering: a partition-and-group framework. In: Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, SIGMOD 2007, pp. 593–604. ACM, New York (2007)
11. Li, X., Han, J., Lee, J.-G., Gonzalez, H.: Traffic density-based discovery of hot routes in road networks. In: Papadias, D., Zhang, D., Kollios, G. (eds.) SSTD 2007. LNCS, vol. 4605, pp. 441–459. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73540-3_25
12. Pelekis, N., Theodoridis, Y.: Mobility data mining and knowledge discovery. In: Pelekis, N., Theodoridis, Y. (eds.) Mobility Data Management and Exploration, pp. 143–167. Springer, New York (2014). https://doi.org/10.1007/978-1-4939-0392-4_7
13. Suykens, J.A., Vandewalle, J.: Least squares support vector machine classifiers. Neural Process. Lett. **9**(3), 293–300 (1999)
14. Terroso-Sáenz, F., Cuenca-Jara, J., González-Vidal, A., Skarmeta, A.F.: Human mobility prediction based on social media with complex event processing. Int. J. Distrib. Sens. Netw. **12**(9), 5836392 (2016)
15. Thomee, B., Shamma, D.A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.J.: YFCC100M: the new data in multimedia research. Commun. ACM **59**(2), 64–73 (2016)
16. Zheng, Y.: Trajectory data mining: an overview. ACM Trans. Intell. Syst. Technol. **6**(3), 29:1–29:41 (2015)
17. Zheng, Y., Li, Q., Chen, Y., Xie, X., Ma, W.Y.: Understanding mobility based on GPS data. In: Proceedings of the 10th International Conference on Ubiquitous Computing, UbiComp 2008, pp. 312–321. ACM, New York (2008)
18. Zheng, Y., Xie, X.: Learning travel recommendations from user-generated GPS traces. ACM Trans. Intell. Syst. Technol. **2**(1), 2:1–2:29 (2011)
19. Zhu, Y., Zheng, Y., Zhang, L., Santani, D., Xie, X., Yang, Q.: Inferring taxi status using GPS trajectories. Technical report, May 2012. arXiv:1205.4378

Check for updates

A Cooperative Multi-Agent System for Wind Power Forecasting

Tanguy Esteoule[1,2(✉)], Alexandre Perles[1], Carole Bernon[1], Marie-Pierre Gleizes[1], and Morgane Barthod[2]

[1] IRIT, University of Toulouse, Toulouse, France
{tanguy.esteoule,alexandre.perles,carole.bernon,marie-pierre.gleizes}@irit.fr
[2] meteo*swift, Toulouse, France
morgane.barthod@meteoswift.com

Abstract. In the coming years, ensuring the electricity supply will be one of the most important world challenges. Renewable energies, in particular wind energy, are an alternative to non-sustainable resources thanks to their almost unlimited supply. However, the chaotic nature and the variability of the wind represent a significant barrier to a large-scale development of this energy. Consequently, providing accurate wind power forecasts is a crucial challenge. This paper presents AMAWind, a multi-agent system dedicated to wind power forecasting based on a cooperative approach. Each agent corresponds to a turbine at a given hour, it starts from an initial production forecast and acts in a cooperative way with its neighbors to find an equilibrium on conflicting values. An assessment of this approach was carried out on data coming from a real wind farm.

Keywords: Multi-Agent System · Cooperation · Wind energy Forecasting

1 Introduction

Renewable energies are increasing steadily in several countries, in particular wind energy, mainly driven by the cost decrease of wind turbines. In 2017, wind energy represented 5% of the French national electricity production with 12 GW installed. The United Nations Conference on Climate Change COP21 has set a goal of 30% renewable energy in the overall energy supply in the country by 2020, and more precisely, the wind installed capacity should reach 26 GW by 2023 [24].

In a number of countries with significant wind power generation, electricity markets are organized as electricity pools, gathering production and consumption offers in order to dynamically find the quantities and prices for electricity generation and consumption maximizing social welfare. Wind power producers propose energy offers based on forecasts. The market clearing is designed to

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 152–163, 2018.
https://doi.org/10.1007/978-3-319-94580-4_12

match production offers and consumption bids through an auction process. Since power producers are financially responsible for any deviation from these contracts, improving wind power forecasting accuracy enables to reduce the penalties they incur [20].

Wind power forecasts have been used industrially for over 20 years and this field is approaching technological maturity following a concerted research effort reviewed comprehensively in [8,16]. However, forecast errors are still high and there are several pointers to improve them. This paper focuses on one of the current challenges introduced by the research and industry stakeholders of the International Energy Agency (IEA) Wind task 36: Model the interactions between the wind turbines of a farm [9].

Indeed, the dependencies between the productions of close turbines represent additional information rarely integrated in the wind power forecasting models. In this paper we propose to solve this problem with a cooperative approach which has shown significant results in other applications [2,22].

This paper presents therefore an approach based on Multi-Agent Systems and cooperation to provide wind power forecasts compliant with the farm constraints. It is organized as follows: Sect. 2 introduces the wind power forecasting problem. The Adaptive Multi-Agent Systems approach is described in Sect. 3 and the resulting designed system called AMAWind (**A**daptive **M**ulti-**A**gent System for **Wind** Power Forecasting) is presented in Sect. 4. Finally, the evaluation of the system and the results analysis are detailed in Sect. 5, before concluding.

2 Wind Power Forecasting

2.1 Theoretical Power Curve

According to theoretical studies on wind turbines [15], the power P delivered by a wind turbine follows the equation:

$$P = \frac{1}{2}\rho S C_p v^3 \tag{1}$$

where v is the wind speed, ρ is the air density, S is the rotor surface (the area swept by the blades) and C_p is the power coefficient (the fraction of wind energy that the wind turbine is able to extract).

This function only forms part of the full power curve (the graph representing the wind speed-production relationship). A typical power curve for an operational wind turbine is sketched in Fig. 1 and is made up of 4 parts: (1) below cut-in wind speed (typically between 3 and 4 m/s) where the turbine does not operate, (2) between cut-in and rated wind speed where the power follows (1), (3) above rated wind speed where the power is limited to the turbine's rated power, (4) above cut-out wind speed (usually around 25 m/s) where the turbine is shut down to prevent damage [4].

2.2 Current Forecast Methods

In practice, the relationship between wind speed and power is difficult to model because the conversion process is affected by many external factors such as poor quality wind speed measurements, mechanical wear and blade erosion, among others. Moreover, the wind speed at the exact blades location depends on the topography and the interaction between turbines. In Fig. 1, the observed production of a wind turbine is also plotted as a function of the 100 m high wind speed forecast, the wide disparity of the points demonstrates the difficulty of modeling the relationship.

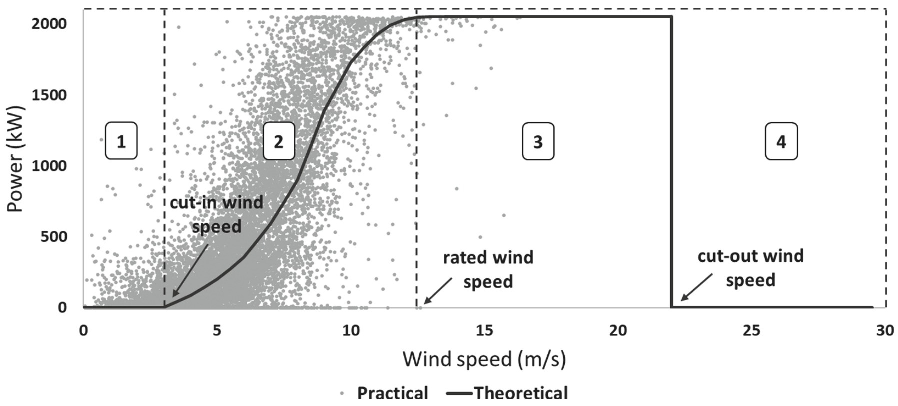


Fig. 1. A theoretical power curve compared with the observed production as a function of the 100 m high wind speed forecast

As a result, power curve models constitute a preliminary approximation of the production but they introduce uncertainty. They are mostly used when a wind speed or production history is not available, e.g. for recently installed turbines.

According to [13], wind power forecasting methods can be basically divided into two distinct categories: (1) Physical approaches based on Computational Fluid Dynamics (CFD) models. (2) Statistical approaches which use previous historical data to train a model representing the relation between wind power and explanatory variables including Numerical Weather Prediction (NWP) and on-line measured data. Hence, either we improve the forecasted wind by modeling the site or we learn from the model errors. Although the first approach is necessary in the absence of measured data on the wind turbine, it is a very specific problem depending on the topography and roughness of the site. Statistical approaches based on machine learning methods such as boosted regression trees [14], neural networks [21] or deep learning [23] are at the forefront of the technology, with gradient boosting methods winning both the 2012 and 2014 Global Energy Forecasting Competitions [11,12]. Despite the performance of these approaches, they do not take into account the information available relating to the relationships between wind turbines.

2.3 Towards a Turbine-Level Forecasting

Due to the low resolution of weather models or a lack of data, wind power forecasts are usually provided at farm scale. In other cases, a production is forecasted for every wind turbine independently and the farm production is obtained by a sum of these forecasts. However, for a single farm, wind turbines productions are very correlated with each other, especially between close turbines. Moreover, since a wind turbine generates electricity from the energy in the wind, the wind leaving the turbine has a lower energy content than the wind arriving in front of the turbine. A wind turbine thus interferes with its neighbors and can cause a production decrease on the turbines located behind it downwind. This phenomenon is called the **wake effect** [17]. This additional information has to be taken into account in the forecast process with the aim of improving the prediction accuracy. Therefore, the problem is to forecast the production at wind farm level by considering local constraints between turbines.

A bottom-up approach has been proposed in [3] to solve this problem. A first forecast is done independently at turbine-level with machine learning algorithms (LASSO, GBM and XGBoost). The farm production is then computed by a weighted sum of the forecasts where weights are determined by linear regression. An improvement can be observed by firstly dividing data by wind direction and then determining weights. However this approach solves the problem in a global way, considering all the turbines together. It does not take into account the local constraints between close turbines.

In this work, a wind farm is considered as a whole system composed of interrelated turbines. Like a mechanical spring system, entities are connected with each other by some constraints. The system evolves towards an equilibrium state globally minimizing the constraints in the system. In this case, an entity tries to comply with the consistency between its forecast and the past observations of its history. Each entity starts from an initial forecast and can modify it in order to make it coherent with the forecasts provided by its neighborhood.

A wind farm is composed of multiple interacting wind turbines within an environment. From this point of view, a Multi-Agent System (MAS) is a suitable system to represent the wind farm structure. Since the relationships between turbines evolve continuously according to weather situations, an adaptive method is required to solve these dynamic constraints. Therefore, the MAS has to be endowed with adaptation capabilities and we focused this study on Adaptive Multi-Agent Systems [5].

3 Adaptive Multi-Agent Systems

An Adaptive Multi-Agent System (AMAS) is a MAS able to adapt to its environment in an autonomous way [5]. In an AMAS, the global function is not hard-coded but emerges from the interactions between the agents composing it. Each agent performs a local partial function and is able to autonomously change its behavior and its relations with the other ones. This ability of self-organization makes the global function evolve and the system becomes able to adapt to disruptions or changes in its environment, it is therefore open, robust and designed in a bottom-up way.

Of course, agents need a criterion to question their behavior and relations, this criterion is what is called "cooperation". Every agent has a cooperative social attitude which makes it always help the most critical agent in its (limited) neighborhood unless itself becomes the most critical one (an agent is benevolent but not altruistic). The criticality of an agent represents its degree of dissatisfaction with respect to its local goal [7]. This domain-dependent criticality value has to be normalized in order to be compared between agents. The actions of the agents aim at minimizing as much as possible the criticality of all the agents in the system without needing any global knowledge.

Despite the fact that this approach is relatively new, recent examples show that these systems can be used to solve complex industrial problems, such as estimation of the state of an electrical network [19], automatic tuning of a combustion engine [2] or optimization of the cooling of photovoltaic panels [10]. Furthermore, the dynamics and uncertainty of data provided as inputs in this wind forecasting problem also confirm the relevance of such a system.

4 AMAWind for Wind Power Forecasting

A weather model provides forecasts on specific coordinates (every 2.5 km horizontally and vertically in our case) called grid points (see Fig. 2) and a wind turbine is directly surrounded by four grid points. Forecasts have to be made considering weather forecast data coming from weather models and production data obtained from the wind turbines of the farm. In this paper, we consider a grid point as the weather forecast database of a coordinate and a wind turbine as the production database of itself. The uncertainty of the weather forecasts, the complexity of the wind speed-production relationship and the interdependence of the wind turbines productions are such that the use of an Adaptive Multi-Agent System is a good candidate to solve the wind turbine production forecast problem.

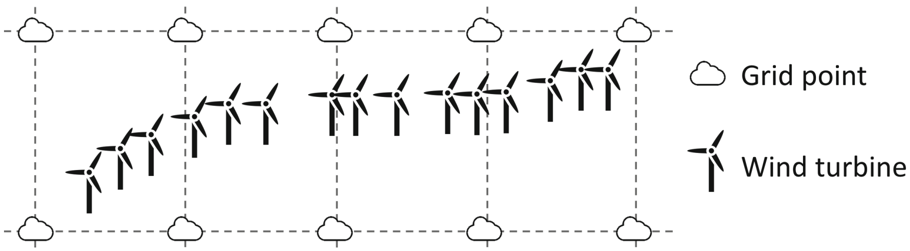


Fig. 2. The layout of the studied wind farm. The weather forecasts are provided at grid points position.

The proposed forecasting system, called AMAWind, is therefore based on an Adaptive Multi-Agent System and was designed in a bottom-up manner by first determining the entities and agents composing this system according to the ADELFE methodology [1].

An active entity, named Grid Point Hour (GPH) entity, is associated with one grid point and one hour and provides weather forecasts for its grid point for the given hour.

An agent, named Wind Turbine Hour (WTH) agent, is responsible of the forecast of a wind turbine production at a given hour (e.g. an agent charged with forecasting for the wind turbine #4 on January 25, 2018 at 08:00 a.m.). The environment of a WTH agent is based on physical closeness: at a given hour, a WTH agent is related to the four closest GPH entities and, at most, the two closest WTH agents (see Fig. 3).

WTH agents and GPH entities have respectively access to their associated wind turbine and grid point history.

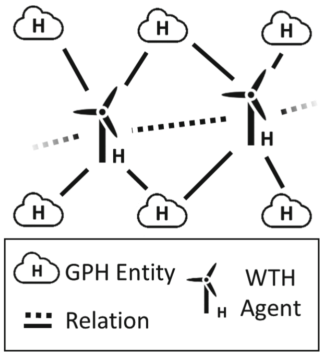


Fig. 3. Agentification

4.1 Agents Behavior

The behavior of an agent follows the classical **Perception-Decision-Action** life cycle. The agent starts with a forecast initialized at the average production of the wind turbine. In each cycle, it can decide to modify it.

Firstly, the agent perceives the current forecast and criticality of its neighbors. It also knows its own forecast and criticality.

Secondly, the agent decides how to change its forecast. These changes are made from the information previously perceived. The agent has three possibilities: to increase, to decrease or not to change its forecast. The increment is fixed at 10 kW. It simulates these different cases and performs the action minimizing the maximal criticality of its neighbors and itself. Even if this action has as a consequence a higher criticality for itself, an agent acts in a cooperative way as described in Sect. 3.

Finally, the agent performs the decided action which possibly changes its forecast, and consequently updates its criticality.

4.2 Criticality

As seen in Sect. 3, the cooperation between agents is mainly based on the concept of criticality. Generally, a criticality C depending on x follows the function:

$$C(x) = \begin{cases} 1 - e^{-k(x-i_1)}, & \text{if } x < i_1 \\ 0, & \text{if } x \in [i_1, i_2] \\ 1 - e^{k(x-i_2)}, & \text{if } x > i_2 \end{cases} \tag{2}$$

where k is a slope factor and $[i_1, i_2]$ is the interval corresponding to the lowest criticality.

The local aim of a Wind Turbine Hour agent is to determine the best forecast and to ensure that this one matches with the ones of its neighbors. Therefore, a high criticality means the forecast does not comply with the constraints imposed by the history. These constraints have to appear in the criticality through the interval $[i_1, i_2]$ chosen in (2). The general process to build the interval from a current weather forecast provided by a Grid Point Hour entity is:

– The weather forecast is compared with the full weather forecasts history of the grid point by means of a Fixed-radius near neighbors search [6]. It consists in looking for the closest entries below a defined threshold according to a similarity measure. In brief, we are looking for the dates where similar weather situations have occurred (e.g. a cold and windy day).
– The powers delivered by the wind turbine corresponding to these dates are extracted from its production history. A set of potential wind power forecasts is thus obtained.
– Finally, the interval is built by selecting the first and third quartiles of this set.

The choice to use an interval simplifies the criticality construction. In the long term, criticality should really represent the distribution of the set of potential wind power forecasts.

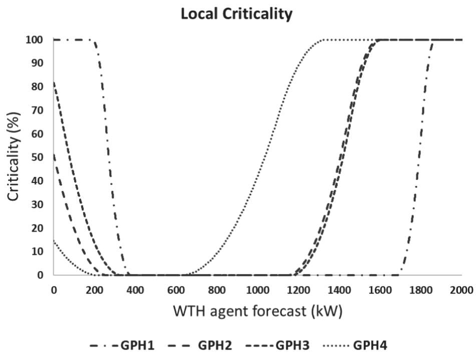
In this paper, the weather situation corresponds to the forecast of the wind speed and the wind direction at a given date. Other indicators indirectly involved in (1) such as temperature, pressure or relative humidity are not used in this work because they have a lesser impact on the production.

The quality of the forecast made by a WTH agent depends on the consistency of its forecast with both its own past productions and the neighboring agents forecast. The criticality is then expressed by considering these two factors and combines two kinds of sub-criticalities (see Fig. 4a and b):

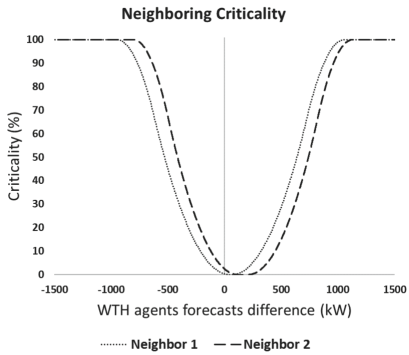
Local Criticality. The forecast made by an agent has to be consistent with the productions observed with a similar weather situation in its history (e.g. the wind turbine rarely produces energy when the weather model forecasts a very light wind). For every pair of WTH agent and GPH entity, an interval is made from the above process. The corresponding criticality is then built with the function (2).

Neighboring Criticality. The forecast has also to be consistent with the ing agents forecasts (e.g. if a wind turbine always produces more amount of power than its neighbor, this constraint has to be taken into account). Indeed, a turbine slows down the wind behind it due to the wake effect and thus interferes with its neighbors. Therefore, a difference appearing in past observations between two wind turbines will be included in the resolution through this criticality. It is built in the same way that the local criticality except that the interval comes from a set of production differences between two wind turbines instead of a set of productions.

The final criticality of a WTH agent corresponds to the maximum between each local criticality and neighboring criticality. This choice enables not to give an advantage to one criticality over the other, they are considered equivalent.



(a)



(b)

Fig. 4. Criticality decomposition: local (a) and neighboring (b)

5 Evaluation

The system was implemented and tested on real data coming from a French wind farm. Given that the main goal of this work is to improve forecast quality, we mainly evaluate the system on forecast error.

5.1 Protocol and Data Set

To evaluate forecasts performance two metrics were used: the mean absolute error (MAE) and the root mean squared error (RMSE), two standard measures to evaluate estimator performance. RMSE is an interesting measure in this use case because larger errors have a disproportionately large effect.

These metrics are given by the equations:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\widehat{y}_i - y_i| \qquad \mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\widehat{y}_i - y_i)^2} \tag{3}$$

where $\widehat{y}_i$ is the forecast and y_i the observation. The NMAE and the NRMSE correspond to the errors normalized by the rated power to have more convenient results.

The wind farm includes 15 lined up turbines. Weather forecasts are provided by the Météo-France AROME high-resolution forecast model from 12:00 a.m. to

06:00 p.m. for the next day (with time horizon from 21 h to 39 h). The experiment covers a large period thanks to a 26-months history of wind power and weather forecasts from 05/2015 to 07/2017.

The model is validated by a k-fold cross-validation. The original sample is partitioned into k equal sized subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k - 1 subsamples are used as training data. Although this method does not represent the real conditions (learning from more recent data than the validated ones), it enables to give an insight on how the model will generalize to an independent data set. In this case, we chose a 7-fold which corresponds to a validation set of approximately 100 days. At each validation, approximately 28500 agents are created (100 days $\times$ 19 h $\times$ 15 turbines).

The forecasts provided by the agents evolve in the system as long as the overall criticality does not converge towards a value. In practical, we confirm the convergence if the criticality remains constant for 10 successive cycles. When this occurs, the wind power forecasts are extracted and compared to real productions.

5.2 Results and Error Comparison

Table 1 presents the NMAE and the NRMSE computed for the 7 subsamples of the 7-fold validation. The average error is then provided at the end.

Table 1. Results summary

Sample	Forecast method					
	Average production		GBM		AMAWind	
	NMAE	NRMSE	NMAE	NRMSE	NMAE	NRMSE
1	20.66%	23.96%	10.23%	15.62%	10.17%	15.53%
2	22.84%	27.09%	10.78%	17.21%	10.48%	16.78%
3	30.75%	37.94%	14.16%	19.98%	14.13%	19.87%
4	23.73%	29.32%	13.19%	19.39%	13.46%	19.90%
5	18.67%	20.67%	7.27%	11.43%	7.00%	10.73%
6	29.05%	35.36%	14.34%	21.75%	14.23%	21.56%
7	20.09%	23.34%	9.30%	14.18%	9.09%	13.66%
Mean	**23.68%**	**28.24%**	**11.32%**	**17.08%**	**11.22%**	**16.86%**

The *Average production* method consists in using the average production of the turbine as a forecast. This naive method, never used in practice, serves as a reference value. In this experiment, it provides an average NMAE of **23.68%**, twice the errors obtained by the two methods based on weather forecasts.

The *GBM* method (Gradient Boosting Model) is a forecaster commonly chosen in wind power forecasting because it provides correct results without requiring complex tuning. The model was trained and tested on the same data sets and was implemented thanks to the machine learning library Scikit-learn [18].

The cooperative method used in AMAWind provides average results slightly better than the GBM in terms of NMAE and NRMSE (respectively from **11.32%** to **11.22%** and from **17.08%** to **16.86%**). Except for the fourth subsample, the forecasts made by the cooperative approach are more accurate.

Given that the wind power forecast error is highly correlated with the weather forecast error, we observe similar results between methods for the same subsample. The wide disparities between the subsamples (e.g. **7.27%** and **14.34%**) are due to the non-linearity of the wind speed and production relationship. Indeed, as can be seen in Fig. 1, a wind speed deviation before or after the cut-in wind speed will not lead to the same error. Therefore, the sixth subsample depicted a more windy period than the fifth.

5.3 Criticality Analysis

Does a criticality decrease lead to an error decrease? This relation, essential to improve the results with such a system, is not directly implemented. The criticality and behavior definitions have to result to this emerging phenomenon.

The evolution of the average NMAE and criticality for the seven subsamples is presented in Fig. 5a and b which show that the required relation is confirmed by the experiment.

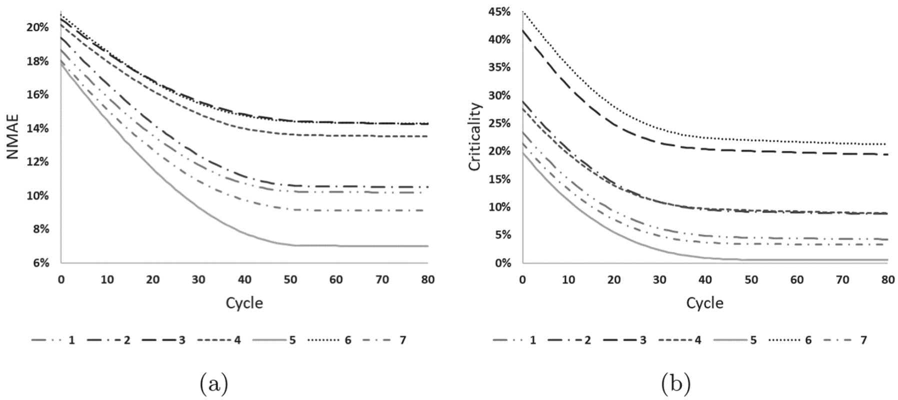


Fig. 5. NMAE (a) and criticality (b) evolution on the seven subsamples

While every criticality decreases in the same way, the errors decrease with different slopes (e.g. the fifth and the seventh). However, for a given subsample, the convergence happens simultaneously for the NMAE and the criticality because an agent would not change its forecast if its criticality reaches a minimum. Moreover, a high final criticality leads to a high final error, e.g. the third and the sixth end at a criticality and error of approximately 20% and 14%. The criticality can therefore be seen as an uncertainty indicator.

In conclusion, the criticality chosen in this work is consistent for the problem because it enables to improve the forecasts.

6 Conclusion

This paper has proposed a wind power forecasting approach based on a cooperative resolution by a Multi-Agent System. We considered the wind farm as a whole system with local constraints between each turbine. We used cooperation as a means to solve the possible conflicts and to obtain a forecast compliant with these constraints.

Resolutions were made locally and showed a decrease of the global forecast error. Despite the basic behavior of the agents, AMAWind provides results slightly better than a commonly used method. These initial outcomes are encouraging but some improvements can be made to the criticality definition and the algorithm used in the initialization.

The future works should take into account temporal dependencies by connecting agents representing the same turbine one hour before and after. Whereas this work focuses on day-ahead forecasting, another potential approach concerns short-term forecasting for the intraday market where temporal dependencies are important.

Acknowledgments. This work is part of the research project Meteo*Swift funded by the ERDF (European Regional Development Fund) of the European Union and the French Occitanie Region and supported by the ANRT (French National Association for Research and Technology). We would also like to thank the CNRM (French Weather Research Centre), our partner in this project.

References

1. Bernon, C., Gleizes, M.-P., Peyruqueou, S., Picard, G.: ADELFE: a methodology for adaptive multi-agent systems engineering. In: Petta, P., Tolksdorf, R., Zambonelli, F. (eds.) ESAW 2002. LNCS (LNAI), vol. 2577, pp. 156–169. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39173-8_12
2. Boes, J., Migeon, F.: Self-organizing multi-agent systems for the control of complex systems. J. Syst. Softw. **134**, 12–28 (2017)
3. Browell, J., Gilbert, C., McMillan, D.: Use of turbine-level data for improved wind power forecasting. In: 2017 IEEE Manchester PowerTech, pp. 1–6 (2017)
4. Burton, T., Jenkins, N., Sharpe, D., Bossanyi, E.: Wind Energy Handbook. Wiley, Hoboken (2011)
5. Capera, D., Georgé, J.P., Gleizes, M.P., Glize, P.: The AMAS theory for complex problem solving based on self-organizing cooperative agents. In: 12th IEEE International Workshops on Enabling Technologies, Infrastructure for Collaborative Enterprises, Linz, Austria, pp. 383–388. IEEE Computer Society (2003)
6. Dickerson, M.T., Drysdale, R.S.: Fixed-radius near neighbors search algorithms for points and segments. Inf. Process. Lett. **35**(5), 269–273 (1990)
7. Georgé, J.P., Gleizes, M.P., Camps, V.: Cooperation. In: Di Marzo Serugendo, G., Gleizes, M.P., Karageorgos, A. (eds.) Self-organising Software: From Natural to Artificial Adaptation, pp. 193–226. Springer, Cham (2011). https://doi.org/10.1007/978-3-642-17348-6

8. Giebel, G., Brownsword, R., Kariniotakis, G., Denhard, M., Draxl, C.: The state-of-the-art in short-term prediction of wind power: a literature overview. Technical report, ANEMOS.plus (2011)
9. Giebel, G., Cline, J., Frank, H., Shaw, W., Pinson, P., Hodge, B.M., Kariniotakis, G., Madsen, J., Möhrlen, C.: Wind power forecasting: IEA Wind Task 36 & future research issues. J. Phys.: Conf. Ser. **753**, 032042 (2016)
10. Guivarch, V., Bernon, C., Gleizes, M.P.: Power optimization by cooling photovoltaic plants as a dynamic self-adaptive regulation problem. In: International Conference on Agents and Artificial Intelligence (ICAART), vol. 1, pp. 276–281. SciTePress (2018)
11. Hong, T., Pinson, P., Fan, S.: Global Energy Forecasting Competition 2012 (2014)
12. Hong, T., Pinson, P., Fan, S., Zareipour, H., Troccoli, A., Hyndman, R.J.: Probabilistic Energy Forecasting: Global Energy Forecasting Competition 2014 and Beyond (2016)
13. Jung, J., Broadwater, R.P.: Current status and future advances for wind speed and power forecasting. Renew. Sustain. Energy Rev. **31**, 762–777 (2014)
14. Landry, M., Erlinger, T.P., Patschke, D., Varrichio, C.: Probabilistic gradient boosting machines for GEFCom2014 wind forecasting. Int. J. Forecast. **32**(3), 1061–1066 (2016)
15. Lydia, M., Kumar, S.S., Selvakumar, A.I., Kumar, G.E.P.: A comprehensive review on wind turbine power curve modeling techniques. Renew. Sustain. Energy Rev. **30**, 452–460 (2014)
16. Monteiro, C., Bessa, R., Miranda, V., Botterud, A., Wang, J., Conzelmann, G., et al.: Wind power forecasting: state-of-the-art 2009. Technical report, Argonne National Laboratory (ANL) (2009)
17. Nygaard, N.G.: Wakes in very large wind farms and the effect of neighbouring wind farms. J. Phys. Conf. Ser. **524**(1), 012162 (2014)
18. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**(10), 2825–2830 (2011)
19. Perles, A.: An adaptive multi-agent system for the distribution of intelligence in electrical distribution networks: state estimation. Ph.D. thesis, Université de Toulouse (2017)
20. Pinson, P.: Wind energy: forecasting challenges for its operational management. Stat. Sci. **28**(4), 564–585 (2013)
21. Ramasamy, P., Chandel, S., Yadav, A.K.: Wind speed prediction in the mountainous region of India using an artificial neural network model. Renew. Energy **80**, 338–347 (2015)
22. Verstaevel, N.: Self-organization of robotic devices through demonstrations. Ph.D. thesis, Université de Toulouse (2016)
23. Wang, H., Li, G., Wang, G., Peng, J., Jiang, H., Liu, Y.: Deep learning based ensemble approach for probabilistic wind power forecasting. Appl. Energy **188**, 56–70 (2017)
24. Wind Observatory: Analysis of the wind power market, wind jobs and future of the wind industry in France. Technical report (2017)

Check for updates

Cooperative Agents for Discovering Pareto-Optimal Classifiers Under Dynamic Costs

Svitlana Galeshchuk[1(✉)] and Sumitra Mukherjee[2]

[1] Paris Dauphine University, Paris 75016, France
svitlana.galeshchuk@dauphine.fr
[2] Nova Southeaster University, Fort Lauderdale, FL 33315, USA
sumitra@nova.edu

Abstract. In contrast to passive classifiers that use all available input feature values to assign class labels to instances, *active* classifiers determine the features on which to base the classification. Motivated by the tradeoff between the cost of classification errors and the cost of obtaining additional information, active classifiers are widely used for diagnostic applications in domains such as in medicine, engineering, finance, and natural language processing. This paper extends the extant literature on active classifiers to applications where cost of obtaining additional information may vary over instances to be classified and over time. We show that this entails training a set of classifiers that grows exponentially with the number of features and propose an efficient way to discover models in the cost-accuracy Pareto optimal frontier. Our method is based on a set of cooperative agents. The incremental contributions of agents to a coalition is used as a surrogate measure to guide a heuristic search for models. Empirical results based on controlled experiments indicate that our approach can identify Pareto-optimal active classifiers under dynamic costs even in domains that involve a large number of input features.

Keywords: Cooperative agents · Feature importance · Heuristic search
Model selection · Pareto-optimal models

1 Introduction

Classifiers trained by supervised machine learning algorithms are widely used to assign labels to instances described by input features. Most such classifiers are *passive* in the sense that they use all available input feature values to assign class labels to instances. In various domains such as in medicine, engineering, finance, and natural language processing, there are costs associated with obtaining input features values. For example, in medical diagnosis, results of diagnostic tests that serve as input features for classification come at a cost.

Active classifiers, in contrast, determine the features on which to base the classification [1]. The expected cost of classification is the sum of the cost of classification errors and the cost of obtaining additional input feature values. Typically, classification errors decrease as the number input features increase. Active classifiers are designed to

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 164–174, 2018.
https://doi.org/10.1007/978-3-319-94580-4_13

exploit the tradeoff between the cost of classification errors and the cost of obtaining additional information to minimize total expected costs.

The extant literature on active classifiers focuses on applications where the costs associated with features are known and fixed. In many application domains the costs associated with features may vary widely not only over instances to be classified, but also over time [2]. For example, costs of medical diagnostic tests differ from region to region, between facilities in the same region, and for patients covered under differing heath care coverage plans. Under such circumstances, since costs cannot be taken into consideration at the time of training the classifiers, the only recourse is to train and maintain a large set of classifiers. Given costs associated with the input features, an appropriate classifier with a low cost of obtaining input feature values that meets desired classification accuracy requirements may then be selected from this set.

Specifically, given a classification task with n possible input features and desired minimum accuracy requirements, all models based on feature subsets that meet the specified accuracy thresholds are identified; we call such models *acceptable*. Given feature acquisition costs for a specific instance to be classified, the cost associated with an acceptable model may then be trivially determined. To capture the tradeoff between the costs for feature acquisition and classification error, we present those acceptable models that are on the Pareto-optimal frontier. Since it is computationally infeasible to train classifiers on all $2^n - 1$ subsets of features as the number of features n increases, we propose a heuristic search strategy that efficiently identifies all acceptable models.

Our search strategy is based on the observation that a model that is based on subset of features of an unacceptable model is also unacceptable. We use this reasonable assumption to prune our search graph in the process of identifying all acceptable models. Nodes are represented by the feature set used to train and test the model. The root node is represented by the full feature set F. Successor nodes are generated by removing one feature at a time. All non-terminal nodes are acceptable classifiers.

Our hypothesis is that a judicious choice of the order in which features are removed can reduce the total number of classifiers to be trained and tested, thus reducing the total time needed to identify the set of acceptable classifiers. Since the removal of a more important feature has a more significant adverse effect on the classification accuracy of the resulting model, the total number of nodes in our search graph may be reduced by considering nodes obtained by the removal of more important features before those obtained by the removing relatively less important features. We empirically investigate this hypothesis through controlled experiments.

A key to the success of this strategy is the measure used to assess the relative importance of features in a classifier. While common supervised machine learning methods provide a ranking of features based on relative scores, these scores do not explicitly consider the dependence among the features. We posit that these dependencies play a critical role in the performance of classifiers. To capture these dependencies, we represent features as agents in a cooperative game and rank features based on a measure that derives from the Shapley values for the agents. This approach is motivated by the work of Fragnelli and Moretti that takes a game theoretic approach for classification of gene expression data [4]. Experimental results bear out our hypothesis that our Shapley value based feature importance measure results in the training and testing of fewer classifiers than one based on alternative feature ranking measures.

Our main contributions may be summarized as follows: First, we develop a computationally feasible heuristic search strategy for identifying acceptable active classifiers. Secondly, we devise a method for ranking features based on Shapley values of agents in a cooperative game that significantly reduces the number of models to be trained and tested in the process of identifying these classifiers. Finally, we demonstrate how Pareto-optimal active classifiers in the cost-accuracy frontier may be selected using our approach.

The rest of the paper is organized as follows: Sect. 2 defines our problem formally. Related work is discussed in Sect. 3. Section 4 describes our methodology. Results are presented in Sect. 5. Section 6 concludes with some observations on our findings and identifies avenues of future research.

2 Active Classifier Selection Problem

The active classifier selection problem may be defined as follows: For a classifier $C^{F,m}$ trained on a set of input features F using supervised machine learning method m, let $\alpha_j(C^{F,m})$ be the expected accuracy of $C^{F,m}$ with respect to some accuracy measure j as determined by testing the trained model. Given minimum acceptable accuracy thresholds λ_j for accuracy measures $j \in \mathcal{A}$, $C^{F,m}$ is said to be *acceptable* with respect to $\mathcal{A}$ if $\alpha_j(C^{F,m}) \geq \lambda_j \, \forall j \in \mathcal{A}$. Given an *acceptable* classifier $C^{F,m}$ with respect to minimum acceptable accuracy thresholds $\lambda_j \forall j \in \mathcal{A}$, our main goal is to identify all subsets $F_s \subset F$ such that $C^{F_s,m}$ is also acceptable.

Training and testing classifiers take a significant computational effort. Since the size $|F|$ of the feature set may be large, it may be computationally infeasible to train and test classifiers using all $2^{|F|} - 1$ subsets of features. We make a reasonable assumption to make the problem tractable: If a classifier $C^{F_s,m}$ is not acceptable, then all classifiers trained on a subset of its feature set F_s are also not acceptable.

Under this assumption, graph search may be used to identify all subsets of F that result in acceptable classifiers. Nodes are represented by the feature set used to train and test the model. The root node is represented by the full feature set F. For a node represented by feature set F_s, a set of $|F_s|$ successor nodes is obtained as $\{F_s - \{f\} | f \in F_s\}$ by removing one feature at a time. Terminal nodes are acceptable if and only if all successor nodes have already been explored; all non-terminal nodes are acceptable classifiers.

A secondary goal of this study is to investigate whether a judicious choice of the order in which features are removed can reduce the total number of classifiers to be trained and tested, thus reducing the total time needed to identify the set of acceptable classifiers. The relative importance of features in a classifier may be estimated. Our hypothesis is that the total number of nodes in our search graph may be reduced by considering a successor node obtained by the removal of a relatively more important feature before a successor node obtained by the removal of a relatively less important feature. We investigate this hypothesis by removing features in different orders to generate successor nodes.

A tertiary goal of this dissertation is to demonstrate the practical benefits of identifying all acceptable models. For each acceptable classifier, its accuracy profile is presented in terms of all accuracy measures $j \in \mathcal{A}$. The cost associated with applying a trained classifier $C^{F,m}$ may be taken to be the sum of the cost of obtaining values for its input feature set F. Let c_i be the cost of obtaining a sample value for feature i. Then the cost of applying an acceptable classifier with feature set F_s for diagnosis is $\sum_{i \in F_s} c_i$. Using these costs, a non-dominated set of classifiers in the Pareto-optimal frontier of the cost-accuracy space may be obtained. Decision makers may make informed decisions regarding the test results to obtain by selecting a subset of features used by some model in this frontier.

3 Related Work

Our work is motivated by the literature on cost-sensitive active classifiers. Greiner et al. [1] address the problem of learning optimal active classifiers. Under certain restrictions they present a probably-approximately correct algorithm for learning such models. Interested readers are referred to [1] for the literature on active classifiers. Our work differs from this body of literature in that it assumes dynamic costs. Since costs are not known a-priori, they cannot be taken into account when training the models. This necessitates the heuristic search based approach that we propose.

The success of our heuristic search strategy depends critically on the order in which features are removed from models. A comprehensive review of feature selection methods is presented in [5]. Feature selection methods may be classified into wrapper, filter, and embedded methods. Wrapper methods treat the classification models as a black box and evaluate the performance of a feature subset based on the performance of a classifier using that set of input features. Filter methods, in contrast, determine the importance of the features independent of classifiers and use these to identify good feature subsets. Embedded methods iteratively identify subsets of good features using a filter approach and assess merits of the subset using a classifier. Our strategy takes such an embedded approach.

Since the efficacy of our search strategy depends on feature ranking, we seek effective ranking methods instead of relying on default feature rankings produced by classifiers. Fragnelli and Moretti [4] show how a game theoretic approach can help with feature selection in the analysis of gene expression data. Their feature scores correspond to Shapley values of players in a cooperative game [7]. Motivated by their work, we too model our agents as participants in a cooperative game and rank features using a measure that is derived from the Shapley values; our measure reflects the contribution of a feature makes to a model.

4 Methodology

In this section we explain our graph search method to identify all acceptable models, present our approach for estimating the relative importance of features, and describe the experimental setting to evaluate our strategy.

4.1 Search Strategy

In our search graph, nodes are models and represented by a pair $\left(F_s, f_p\right)$, where F_s is the set of features used to train and test a model, and f_p is the feature removed from the parent of the node to obtain the model. The root node contains the full feature set F, and is represented as $(F, \emptyset)$. Let $\mathcal{R}(f)$ denote the ranking of feature f based on its relative importance. For a node $\left(F_s, f_p\right)$, a set of $|F_s|$ successor nodes is obtained as $\left\{F_s - \{f\} | f \in F_s \wedge \mathcal{R}(f) > R\left(f_p\right)\right\}$ by removing those features with ranks greater than f_p. For a fixed ranking of features, this ensures that our search graph is a tree and that every node is visited only once. Nodes representing classifiers that are not acceptable are not expanded; all non-terminal nodes are acceptable classifiers. We maintain a priority queue of nodes to be explored and add models to the queue based on the relative importance of the feature removed to obtain the model. The queue is initialized to contain the model with all features and extended by adding successors to the front of the queue in the order of decreasing importance of the feature removed to obtain the successor. Figure 1 presents the pseudo code

```
search(featureSet F, ranking R):
  acceptableModels = [] # acceptable models
  priorityQ = [F] # initialize with root node
  nTrained = 0 # number of models trained
  while priorityQ: # queue not empty
    nTrained += 1 # train another model
    thisModel = priorityQ.pop() # get first model
    if acceptable(thisModel): # acceptable model
      acceptableModels.append(thisModel) # add to result
      priorityQ.extend(successors(thisModel, R)) # extend priorityQ
  return nTrained, acceptableModels

successors(thisModel =(Fs, fp), ranking R):
  models = []
  for f in Fs:
    if R(f) > R(fp): models.append((Fs-{f}, f))
  return models
```

Fig. 1. Search strategy to identify qualifying models.

To see how our search strategy reduces the number of models to be trained and tested when removing features in decreasing order of importance, consider a simple example with 5 features labeled 0–5 with the labels reflecting the feature score. Of the $2^5 - 1$ possible models, let acceptable models be the ones with a sum of feature score exceeding 7. Figure 2 shows the results of our search using 3 different ordering of features. Note that removing features in decreasing order of importance requires fewer models to be trained and tested. This improvement becomes more significant as the number of features increase.

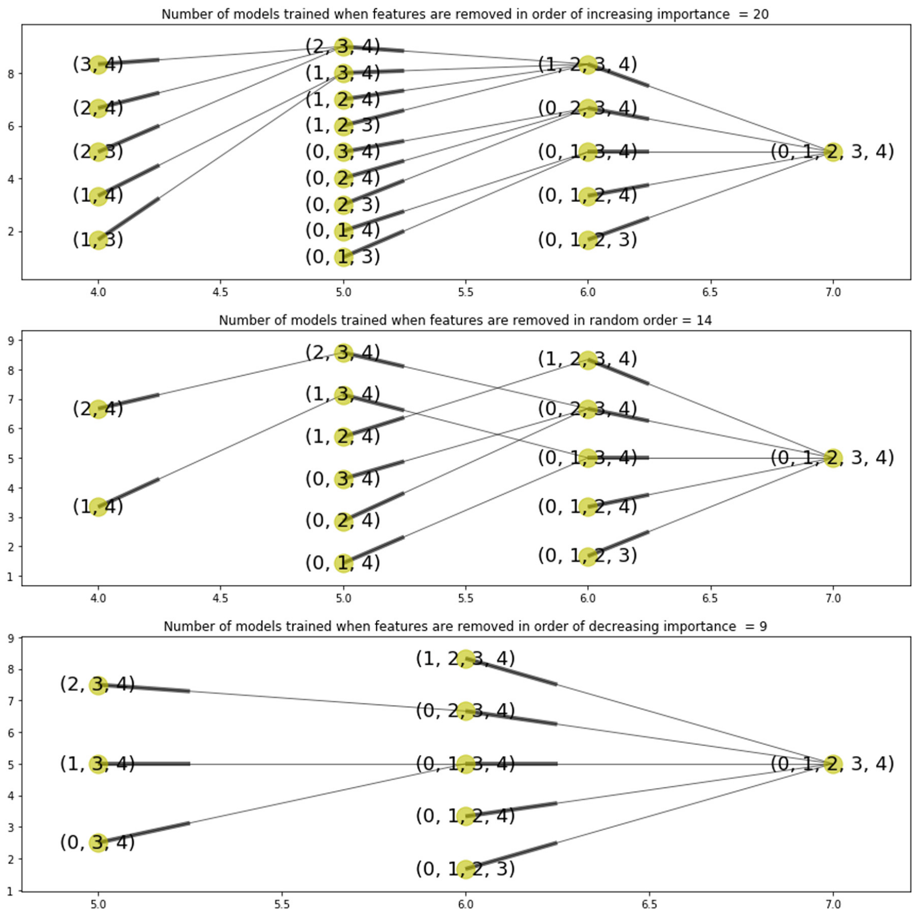


Fig. 2. Removing features in decreasing order of importance requires fewer models to be trained.

4.2 Determining the Relative Importance of Features

For our search strategy to be effective, we need to assess the relative importance of the features accurately. We achieve this by treating the features as agents in a cooperative game. The payoff $v(m)$ associated with agents cooperating to form a model m is taken to be the classification accuracy of that model. We then compute the relative contribution of each agent using a measure based on Shapley value. The Shapley value for an agent i in a cooperative game is given by $\psi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!}(v(S \cup \{i\}) - v(S))$. Computing the payoffs for all possible subsets of F becomes infeasible as the number of features increase. We therefore take an approximate measure based on subsets S of size $|F|-1$ and $|F|-2$. There are $|F| + |F|(|F|-1)/2$ such subsets and hence it is computationally feasible to estimate approximate Shapley values for features even when the number of features is fairly large. While we use this naïve approximation method to compute Shapley values, more sophisticated approximation methods, such as the linear approximation method presented in [3] may be used. To evaluate of feature ranking approach, we compare it to two other ranking methods: the first assigns ranks based on the default feature importance scores produced by the classifier, and the second ranks the features in a random order.

4.3 Experimental Setup to Evaluate Our Method

We use the digit recognition competition problem as our classification task with the data set from https://www.kaggle.com/c/digit-recognizer. The data set consists of 42,000 examples labeled with the correct digit (0–9). Each example contains 714 features: gray scale values (between 0 and 255) for 28 × 28 pixels. The images are rescaled to 7 × 7 pixels representations using the transform module from the Python *scikit-image* library (http://scikit-image.org/docs/dev/api/skimage.transform.html). We use 28,000 randomly selected examples for training and cross-validation and the remaining 14,000 examples for testing. Models are trained and cross-validated on the training set and accuracies of the trained models are evaluated using the test data.

We use a random forest classifier from the Python scikit-learn library (http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html) and train models using an ensemble of 100 trees. Nodes with fewer than 20 samples are not expanded. The accuracy, precision, recall, and F1-score of the trained models with the test set are recorded. Default feature importance scores are those determined by the *RandomForestClassifier* scikit-learn method [6].

We define our acceptable model as follows: Let the set of accuracy measures $\mathcal{A} = \{accuracy, precision, recall, F1score\}$ and let $max(j)$ denote corresponding average score j for the model with all 49 features. Using 5 different values of $\rho \in \{0.99, 0.98, 0.96, 0.96, 0.95\}$, we construct 5 sets of thresholds $\lambda_j = \rho.max(j)$ and deem a model to be acceptable if $\alpha_j(C^{F,m}) \geq \lambda_j \forall j \in \mathcal{A}$. We report our results for the 5 problem instances generated by the 5 sets of thresholds.

Recall that the cost of applying an acceptable classifier with feature set F_s for diagnosis is given by $\sum_{i \in F_s} c_i$, where c_i is the cost of obtaining a sample value for feature i. In our experiments, we generate uniformly distributed costs such that $\sum_{i \in F} c_i = 1$. That is, the cost of acquiring input information for a model is normalized such that the cost of the model with all available input features is 1. An acceptable model with cost c and accuracy α is said to be on the Pareto optimal frontier if and only if all other acceptable models with accuracy $\geq \alpha$ have a cost $> c$, and all other acceptable models with cost $\leq c$ have accuracy $< \alpha$.

5 Results

For the model with all 49 features, we achieve 94% classification accuracy. Indeed our score for all four accuracy measures are 94%. The precision, recall, *F1*-score, and support with the test data of the trained model are reported in Table 1 below. To define acceptable models in our 5 problem instances, we therefore use threshold of 93%, 92%, 91%, 90%, and 89%, respectively, for all four measures.

Table 1. Results with test data of model trained with all 49 features.

Digit	Precision	Recall	F1-score	Support
0	0.96	0.98	0.97	1393
1	0.97	0.97	0.97	1561
2	0.92	0.92	0.92	1451
3	0.92	0.91	0.91	1439
4	0.94	0.93	0.93	1331
5	0.94	0.92	0.93	1265
6	0.95	0.97	0.96	1378
7	0.95	0.94	0.94	1463
8	0.93	0.9	0.91	1330
9	0.9	0.92	0.91	1389
Average	0.94	0.94	0.94	Sum = 14000

Recall that to evaluate the efficacy of our Shapley value based method for feature ranking using cooperative agents, we compare it to two other ranking methods: the first assigns ranks based on the default feature importance scores produced by the classifier, and the second ranks the features in a random order. We evaluate the efficacy of our search strategy as follows: The fewer the number of models trained and tested in the process of identifying acceptable models, the better is the method. Let n_c, n_d, and n_r denote the number of models trained and tested under our cooperative agent based method, the method based on default feature scores, and under random ordering. In Table 2 we present the ratio n_d/n_c and n_r/n_c for each of the 5 problem instances.

Clearly, our cooperative agent based feature ranking method does significantly better than a random rank assignment. Almost twice the number of models get trained and tested under the random assignment than under our method when the accuracy threshold for acceptable models is set at 93%. As the number of acceptable models increase, the advantage of our method becomes more significant. When the accuracy threshold for acceptable models is set at 89%, our approach has over a four-fold advantage. Using the default feature ranking has obvious advantages over random ranking too, but does not perform as well as our method. On the average, it trains and tests about twice as many models as does our method.

Table 2. Ratio of the number of models trained and tested under alternate methods to the number of models trained using our cooperative agent based ranking.

Problem instance	n_d/n_c	n_r/n_c
$\lambda = 0.93$	1.93	1.93
$\lambda = 0.92$	1.96	2.56
$\lambda = 0.91$	2.16	3.19
$\lambda = 0.90$	2.16	3.43
$\lambda = 0.89$	2.19	4.21

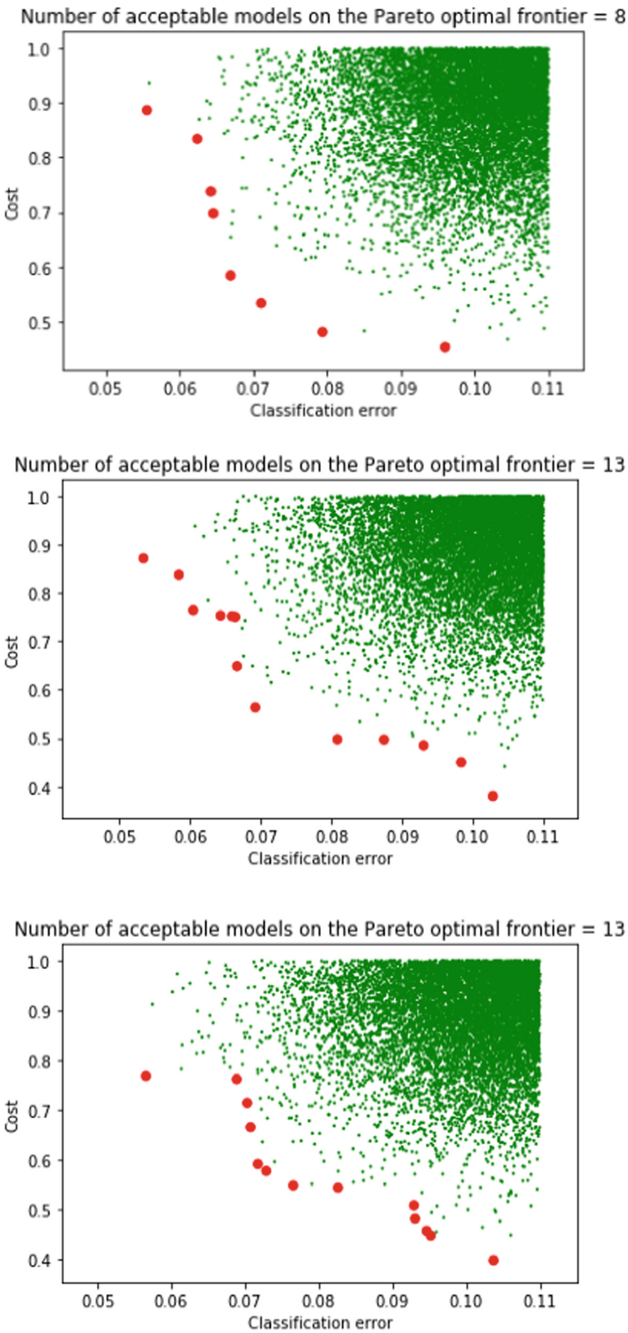


Fig. 3. Acceptable models on the cost-accuracy Pareto optimal frontier are shown in red for three different random cost assignments. (Color figure online)

Finally, to demonstrate how our approach can help practitioners, we identify acceptable models in the Pareto optimal cost-accuracy frontier. Figure 3 shows acceptable models in green and the ones on the frontier in red for the case where acceptable model accuracy threshold is set at 89%. The three sub-figures represent scenarios with three different random assignment of costs to features. Notice that only a very small number of acceptable models are Pareto optimal (8 models in one case and 13 models each in the other two cases). The cognitive load of decision makers may be reduced by presenting them with the cost-accuracy profile of these Pareto optimal models to help them select one that best suits their needs.

6 Conclusions

In this paper we present a computationally feasible heuristic search strategy for identifying classifiers with subsets of features that meet specified accuracy requirements. The efficacy of our search for these acceptable models depends on the order in which features are removed from models until accuracy falls below required thresholds. Thus the ranking of feature importance is critical. While default feature rankings of classifiers help, we demonstrate that our method for ranking features based on Shapley values of agents in a cooperative game further reduces the number of models to be trained and tested in the process of identifying these classifiers. Our work is motivated by applications of active classifiers where the cost of feature acquisition is not known a-priori. Given arbitrary costs, Pareto-optimal active classifiers in the cost-accuracy frontier may be identified using our approach and presented to the decision-makers.

It is feasible to use our approach when the number of features is not too large and only a small fraction of all models meets the desired accuracy standards. The problem of identifying all feasible models in a feature space of size n may be formulated as a search for a maximal antichain of acceptable models in a 2^n lattice. Consider a graph where vertices correspond to feature sets of acceptable models, and there is an edge (x, y) iff $x \subset y$. An antichain is an independent set of nodes in this graph. Since the maximum independent set problem is NP-hard, heuristic methods to infer good antichains is an open research challenge.

References

1. Greiner, F., Grove, A.J., Roth, D.: Learning cost-sensitive active classifiers. Artif. Intell. **139** (2), 137–174 (2002)
2. Whaley, C., Brown, T.T.: Association of reference pricing for diagnostic laboratory testing with changes in patient choices, prices, and total spending for diagnostic tests. JAMA Intern. Med. **176**(9), 1353–1359 (2016)
3. Fatima, S.S., Wooldridge, M., Jennings, N.R.: A linear approximation method for the Shapley value. Artif. Intell. **172**(14), 1673–1699 (2008)
4. Fragnelli, V., Moretti, S.: A game theoretical approach to the classification problem in gene expression data analysis. Comput. Math Appl. **55**(5), 950–959 (2008)
5. Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J., Liu, H.: Feature selection: a data perspective. ACM Comput. Surv. **50**(6), 45 (2017). Article 94

6. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
7. Shapley, L.S.: A value for n-person games. In: Kuhn, H.W., Tucker, A.W. (eds.) Contributions to the Theory of Games. Annals of Mathematical Studies, vol. 28, pp. 307–317. Princeton University Press (1953)

Unemployment Expectations in an Agent-Based Model with Education

Luca Gerotto(✉) and Paolo Pellizzari

Department of Economics, Ca' Foscari University,
Cannaregio 873, 30121 Venice, Italy
{luca.gerotto,paolop}@unive.it

Abstract. Why are unemployment expectations of the "man in the street" markedly different from professional forecasts? We present an agent-based model to explain this deep disconnection using boundedly rational agents with different levels of education. A good fit of empirical data is obtained under the assumptions that there is staggered update of information, agents update episodically their estimate and there is a fraction of households who always and stubbornly forecast that the unemployment is going to raise. The model also sheds light on the role of education and suggests that more educated agents update their information more often and less obstinately fixate on the worst possible forecast.

Keywords: Agent-based modeling · Bounded rationality
Unemployment expectations

1 Introduction

Why are unemployment expectations of the "man in the street" so strikingly different from the ones produced by professional economists or highly regarded institutions? The question is extremely important since the correct unemployment forecast is related to the quantification of the zero-income probability that has been proven to be one of the most important drivers in saving decisions (Carroll et al. [1]).

Several papers have addressed the issue of opinion formation, with a particular focus on contagion among peers, convergence and the dynamics of opinions (e.g., Hegselmann and Krause [2]; for a survey of opinion formation models and possibly related topics, see Castellano et al. [3]). A recent paper by Hegselmann et al. [4] has focused on the role of strategic agents which may state an opinion different from their true one in such a way to influence the ones of other agents. We take a different stand and follow the epidemiological metaphor used in Carroll [5] to study inflation expectations, where the diffusion of information is depicted in pretty much the same way in which epidemics spread: there is no exchange of information among agents yet only some agents are "infected" at time t by a common source (say, mass media coverage) and react accordingly, whereas the ones that were not infected keep using the obsolete information or

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 175–186, 2018.
https://doi.org/10.1007/978-3-319-94580-4_14

act on the basis of spurious facts. We examine survey data of Italian households and show that, while their forecasts are only loosely related to or seemingly disconnected from professional estimates,[1] a model featuring scattered update of information together with the frequent recourse to over-pessimistic forecasts and oblivion is sufficient to accurately replicate the empirical data.

The fact that standard agents produce a vastly different set of forecasts when compared with professional data demonstrates once more how the assumption of perfect rationality of economic agents is flawed even if it is still present in the majority of scholarly work. Indeed, there is overwhelming evidence in experimental economics and psychology that shows the limits of human rationality (Rabin [6]) and how poorly humans behave in problems that involve complex situations (Camerer [7]). Realistic agents were shown to have troubles to gather and correctly internalize a lot of information in a probabilistic setup that is likely to be needed in the proper economic analysis of unemployment. If it is hard to believe that agents are fully rational, it is also puzzling to accept that agents entirely fail to acquire information or are unable to grasp even the most basic elements driving the risk of unemployment: in what follows we assume that households are boundedly rational in the sense that their expectations are often based on inaccurate grounds due to their limited ability or willingness to observe (and absorb!) relevant information. In detail, only the most informed fraction of the population acquires the relevant knowledge and uses a professional forecast. The remaining portion behaves as (very) naive econometricians (Branch [8]), projecting old trends into the future, or stubbornly and pessimistically keeps saying that unemployment is always going to increase in the future. A similar idea has been developed in different setups: Axtell and Epstein [9], for instance, consider a model where a handful of workers are rational in their retirement decisions whereas most of the population just adopts herding strategies imitating the behavior of the peers in the social network; Moro and Pellizzari [10] study the labour market participation where agents mimic (only) the ones with approximately the same age and health who appear to have higher satisfaction (or "utility") from work and leisure.

While in the Agent-Based Models (ABMs) literature on opinion formation most papers involve heterogeneous agents (including Carroll [5]), few works attempt to explicitly relate the heterogeneity of agents with their demographic characteristics. On this perspective, we have included in the model the dependence on the education of the agents, as it is likely that more (less) educated households are more (less) likely to get informed and use professional forecasts.[2] We calibrate the model to empirical data fitting, for each education level, the

---

[1] By professional forecasts in this paper we refer to the figures made public by OECD every quarter. The use of a specific set of "professional" forecasts is not affecting the results sensibly as other forecasts, say produced by different research centers or governmental offices, are typically very similar and strongly correlated with the OECD data.

[2] There are several micro-level empirical studies on this topic, for example Lusardi and Mitchell [11], Souleles [12] and Easaw et al. [13].

probability to obtain the professional forecast as well as the probability to stubbornly declare that unemployment is going to increase.

Interestingly, our results show that empirical households' expectations are quite similar to the ones produced by the simulations. In addition, more educated agents more often get and use professional forecasts and less frequently exhibit stubborn pessimism. In this sense, our work shows a plausible way to reconcile data with a realistic behavior of the agents (who are not perfectly informed and fail to be fully rational at all times).

Our model is "explicit", as suggested in Epstein [14], it is sufficient to replicate the most salient features of the data and, even if we cannot claim it necessarily says the last word, it sheds light on reasons accounting for different forecasts: the fact that households predictions are described by modest rationality and by frequently resorting to over-pessimistic opinions rather than to official data is of undoubted relevance for policy making, in particular as far as this problem is exacerbated for the less educated. Our work is quite evidently related to the generative approach to modelling and offers a tentative explanation as we can "grow" the phenomenon of interest, see Epstein [15].

The paper is organized as follows. Section 2 describes the data we have used and shows how *prima facie* survey observations appear to be very weakly aligned with professional forecasts. The agent-based model using education-dependent types is described in Sect. 3. We then present and discuss the results of a series of NetLogo simulations. Section 5 contains the conclusion and some final remarks.

2 The Data

In this paper we make use of empirical data on households' and professionals' forecasts. The first are gathered by the Italian National Institute of Statistics (ISTAT) on unemployment expectations. Specifically, ISTAT each month asks to a randomly selected sample of 2000 households the following question:

> How do you expect the number of people unemployed in Italy to change over the next 12 months?

Users have the option to provide an answer in the set {increase sharply, increase slightly, remain the same, decrease slightly, decrease sharply, do not know}. Answers are encoded using the values $\{+2, +1, 0, -1, -2, 0\}$, respectively.[3]

Data are aggregated forming the average of the answers and multiplying by 100, effectively weighting twice as much the "sharp" replies with respect to the "slight" ones. Therefore, the balance index has a theoretical range going from -200,[4] if 100% of interviewees expect unemployment to *decrease* sharply in one year, to +200, if 100% of interviewees expect unemployment to *increase*

[3] "Remain the same" and "do not know" are both encoded as zero.

[4] A "balance index" is constructed as the difference between the percentages of respondents giving positive and negative replies.

sharply in one year. Moreover, sub-indexes that provide expectations for particular demographic categories are available. The comparison of the sub-indexes is not a negligible topic, since several empirical works suggest that the degree of financial literacy (Lusardi and Mitchell [11]) and expectations (Souleles [12] and Easaw et al. [13]) are related to the demographic characteristics of the individual, with a relevant role of the education level.

We define as *professional forecasts* the ones presented in the Organisation for Economic Co-operation and Development (OECD) Economic Outlook for Italy, which is released on a biannual basis.[5] In our analysis we use the forecasted change in the unemployment rate, measured as the difference between the forecasted unemployment rate in four quarters and the unemployment rate of the current quarter.

The top panel of Fig. 1 represents the quarterly indexes for the three different education levels, namely *Less than High School, High School, College*. The plot shows that the three series are correlated but have different levels and that the range of values actually attained by the indexes in the last 22 years ($[-25, 115]$) is quite narrow with respect to the theoretical one ($[-200, 200]$). For example, the index for less educated households has never attained negative values meaning that, in this 22 years window, they never expected the unemployment rate to decrease. This is clearly not justified by neither OECD forecasts (Fig. 1, in the bottom panel) nor ex-post realizations, since the 90 quarters we consider are split almost equally between periods of increasing and of decreasing unemployment, and the average value of the change in the unemployment rate is close to zero. Hence, the series of professionals' and households' forecasts have a strikingly different interpretation.

Table 1. Summary statistics (1995Q1–2017Q2)

	Min	1st Qu	Median	Mean	3rd Qu	Max	St.Dev
Less than High School	4.97	35.84	50.4	53.34	64.9	114	24.79
High School	−9.6	24.89	39.73	44.27	58.32	113.7	28.35
College	−23.47	15.77	32.7	39.09	54.32	109.5	32.39

Table 1 shows some descriptive statistics of ISTAT data. It is apparent that less educated individuals are more pessimistic (higher mean and median) and change less frequently their opinion (lower standard deviation). This stylized evidence, implying that the knowledge and use of relevant information (i.e., professional forecasts) differs as a function of the education level, will be incorporated in the model described in the next section.

[5] From 2003, OECD releases forecasts also for each quarter, while for the period from 1995 to 2002 OECD releases forecasts only for each semester. Therefore until 2002 we estimate the missing quarters forecasts through interpolation of the biannual forecasts.

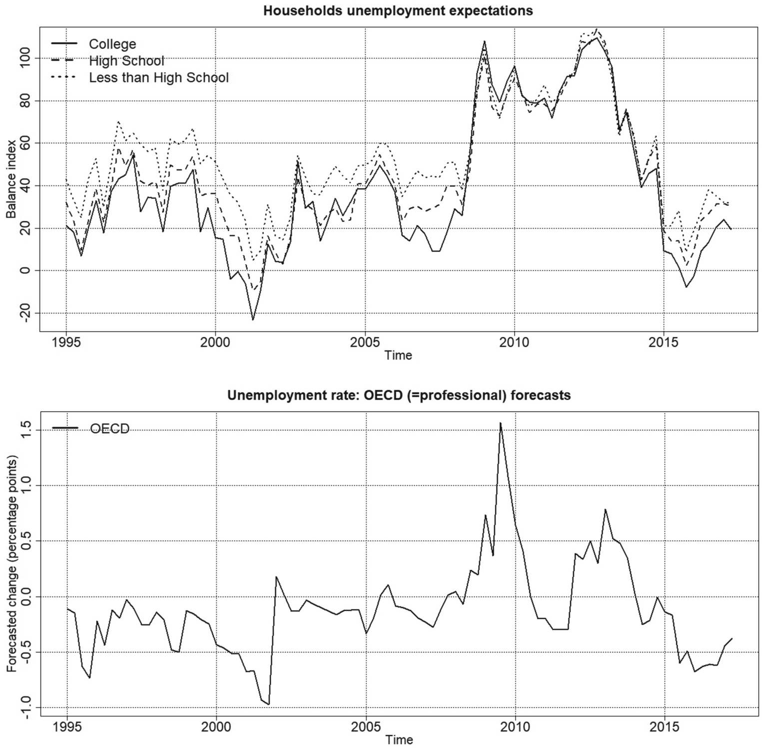


Fig. 1. Above: households' unemployment expectations balance indexes by education level (1995Q1–2017Q2). Below: forecasts of the one-year change in the unemployment rate, as released by the OECD (1995Q1–2017Q2)

3 The Model

Assume there are N agents, whose education level is $edu_i \in E = \{lths, hs, col\}$, with the strings denoting *Less than High School, High School* and *College*, respectively.[6]

At (the beginning of) time t all agents observe a signal s_{it} defined as:

$$s_{it} = \begin{cases} o_t & \text{with probability } \lambda_i \\ s_{i,t-1} & \text{with probability } \beta(1-\lambda_i) \\ 0 & \text{with probability } (1-\beta)(1-\lambda_i) \end{cases}, \tag{1}$$

where o_t is the latest professional forecast released by OECD, $\lambda_i = \lambda(edu_i)$ is the probability to obtain it and β is the probability to remember the past signal if one does not obtain the latest professional forecast. In other words, at each time

[6] 2000 individuals are interviewed by ISTAT per month (6000 individuals per quarter). About $\frac{1}{2}$ of the Italian population belongs to the *lths* group, about $\frac{1}{3}$ to the *hs* group and about $\frac{1}{6}$ to the *col* group. Hence, the baseline simulation has 3000 agents per quarter with *lths*, 2000 with *hs* and 1000 with *col*.

step agents can either get an informative signal o_t, with some probability that depends on their education, or remember the old signal $s_{i,t-1}$ if they get no fresh news, or forget even the old signal. Observe that, with probability $\beta(1-\lambda_i)$, the agent is not informed but keeps using the only item of obsolete information he owned in the past. If, say, $\beta = 0$ then any uninformed agent would pretend that unemployment will not change; at the other extreme, if $\beta = 1$, any uninformed agent would remember and make use of the old signal.

Once the (informative or uninformative) signal is available, agents convert it into an answer ans_{it} to be reported in the survey. Recall from Sect. 2 that answers are encoded with integers in $\{-2, -1, 0, +1, +2\}$ to mean a range going from "unemployment will decrease sharply" (-2) through "unemployment will remain the same" (0) to "unemployment will increase sharply" ($+2$).

Capistrán and Timmermann [16] have shown that, in case of asymmetric losses, agents may prefer to over(under)-predict a variable as this prevents a more costly under(over)-prediction. In our framework, risk-averse agents may prefer to systematically overestimate the unemployment risk for precautionary purposes. We assume that agents stubbornly report $ans_{it} = 2$ with probability $\mu_i = \mu(edu_i)$; otherwise they "translate" their signal based on its perceived magnitude:

$$ans_{it} = \begin{cases} +2 & \text{with probability } \mu_i \\ f(s_{it}|\gamma) & \text{with probability } 1-\mu_i \end{cases}. \tag{2}$$

Each quantitative expectation corresponds to a qualitative opinion and the translation function f depends on a threshold $\gamma > 0$ that shapes 5 possible answers:

$$f(s_{it}|\gamma) = \begin{cases} +2 \text{ if } s_{it} \geq \gamma; \\ +1 \text{ if } \gamma/2 < s_{it} < \gamma; \\ 0 \text{ if } -\gamma/2 \leq s_{it} \leq \gamma/2; \\ -1 \text{ if } -\gamma < s_{it} < -\gamma/2; \\ -2 \text{ if } s_{it} \leq -\gamma. \end{cases} \tag{3}$$

The interpretation is related to the salience of the signal: when it is very large, i.e., exceeding γ, the agent will claim that unemployment will increase sharply. If, instead, the signal is larger than $\gamma/2$ but smaller than γ, the milder conclusion is that "unemployment is going to increase slightly". Finally, if it is close to zero, in the sense that $-\gamma/2 \leq s_{it} \leq \gamma/2$, the answer will state that the rate is not changing (and so on, with obvious modifications for negative values of the signal s_{it}).

At the end of period t, when all $ans_{it}, i = 1, ..., N$, are available, it is straightforward to compute the aggregate index according to the way used by ISTAT:

$$x_t = \frac{100}{N} \sum_{i=1}^{N} ans_{it}.$$

Let y_t denote the balance index and let $y_t^{lths}, y_t^{hs}, y_t^{col}$ be the three subindexes relative to the different levels of education in E. Each run of the model produces a sequence of computed expectations $\{x_t, t = 1, ..., n\}$ and indeed the code can be thought of as an artificial data-generating process of surrogate data that can be compared with the empirically observed expectations $\{y_t, t = 1, ..., n\}$. The differences between the values of x_t and y_t clearly depend on the values of the parameters β and γ, which are constant for all agents, and of parameters λ and μ, which are heterogeneous and affected by the education level (*lths, hs, col*). If $\Theta^{edu} \in [0,1]^2 = (\lambda^{edu}, \mu^{edu})$ denotes the vector of the parameters specific to education level $edu \in E$, we can calibrate the model minimizing the fitting error on the related subindex y_t^{edu}.

There is a wide variety of possible and meaningful fitting criteria and, in order to obtain $\hat{\Theta}^{edu}$, we have used a very simple approach and have computationally minimized the sum of the relative deviations of the first two moments of the time-series x_t (generated by the model) and y_t^{edu} (by ISTAT):

$$\hat{\Theta}^{edu} = \arg\min_{\Theta^{edu}} \left(\left(\frac{E[x_t] - E[y_t^{edu}]}{E[y_t^{edu}]} \right)^2 + \left(\frac{V[x_t] - V[y_t^{edu}]}{V[y_t^{edu}]} \right)^2 \right), \qquad (4)$$

where $E[\cdot]$ and $V[\cdot]$ are the usual mean and variance operators. A list and a brief description of the symbols used in the model are included in Table 2.

It is worth stressing that the model describes a tale in which boundedly rational agents, who seldom acquire and process the relevant data, generate a series of expectations that can be reconnected to some extent with those very same data. Indeed the "man in the street", despite its failure to use professional analysis immediately, makes episodic use of the data and resort to pessimistic forecasts from time to time, based on his level of education.

Table 2. List of symbols and acronyms

Symbol	Description
edu	Education level in the set $E = \{lths, hs, col\}$
o_t	Official professional forecast by OECD
y_t, y_t^{edu}	ISTAT balance index and subindex relative to $edu \in E$
s_{it}	Signal received by agent i at time t
ans_{it}	Survey answer provided by agent i at t
$f(s_{it})$	Translation function
β	Probability of remembering the past signal
λ^{edu}	Probability of absorbing the current official forecast
μ^{edu}	Fraction of stubbornly pessimistic households
γ	Parameter determining the qualitative translation of the signal
Acronym	Meaning
lths	*Less than High School*
hs	*High School*
col	*College*

In a more abstract interpretation, the model is just a filter taking as an input the OECD data and generating the "filtered data" published by ISTAT. Even though purely orthodox economists often expect the two sequences to be the same, this is clearly not the case. The filter, embedding scattered update of information, effects of education and use of bleak forecasts, shows that the observed data can be realistically linked with the professional estimates.

Next section will present the results obtained using the model with calibrated parameters.

4 Results

The model was implemented in NetLogo [17] a powerful and popular computational platform to develop agent-based models in an object-oriented setup with plenty of graphical facilities (download is free at http://ccl.northwestern.edu/netlogo/).[7]

Essentially, the user can change the parameters of the model and obtain a simulated ISTAT time-series starting with the unique inputs provided by OECD every quarter. A screenshot of the model is visible in Fig. 2.

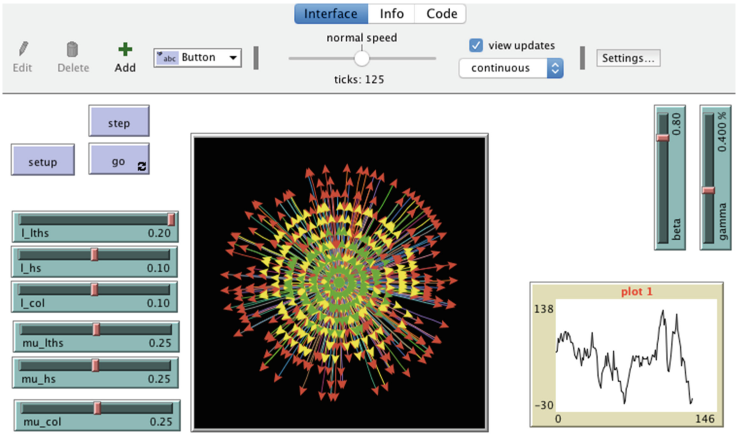


Fig. 2. A screenshot of the NetLogo model. Sliders can change the values of the parameters and the time series of the simulated data appear in the Plot window.

As a preliminary step, we calibrate the two parameters β and γ, which are common among the three education groups, over a grid of possible values. The calibration was obtained using NetLogo's `BehaviorSpace`, a tool allowing the efficient generation of data through the systematic exploration of the parameter space of the model and we picked the values $\beta = 0.9$ and $\gamma = 0.4$. Suppose,

[7] The code can be downloaded at http://virgo.unive.it/paolop/epidemics-istat-paams.nlogo.

for instance, that an agent equipped with his individual μ receives the signal $s_t = o_t = 0.21$ at time t: as the signal is bigger than $\gamma/2$ (but smaller than γ), the answer will be $ans_t = 1$ with probability $1 - \mu$ or $ans_t = 2$ if he acts stubbornly with probability μ. Assume next that the agent fails to get the OECD update o_{t+1}. Then, with probability $\beta = 0.9$ he remembers the past signal: $s_{i,t+1} = s_{it} = 0.21$, so the answer will again be $ans_{t+1} = 1$, or $ans_{t+1} = 2$ if he acts stubbornly. Conversely, with probability $1 - \beta = 0.1$ he does not remember the past signal: in this case $s_{i,t+1}, = 0$ and the answer would be $ans_{t+1} = 0$ (or $ans_{t+1} = 2$ if he acts stubbornly).

Subsequently, we run separate calibrations for each education level, with the aim of obtaining the values of λ^{edu} and μ^{edu} as described by (4). Table 3 displays the calibrated parameters and illustrates how education influences the behavior of the agents. A look at the first column, reporting μ as a function of the education level, shows that the probability to act stubbornly visibly declines for more educated households (for example agents with a college degree have a μ about 20% smaller than their peers who did not complete the high school).

Table 3. Calibrated values of μ, the probability to answer stubbornly, and λ, the probability to acquire the current OECD forecast.

	μ	λ
LTHS	0.309	0.085
HS	0.270	0.095
College	0.251	0.110

As far as the probability to acquire information is concerned, the more (less) educated agents more (less) often are infected by OECD data and the λ's are in the range 0.085 to 0.110 per quarter.[8] Even though the figures may look quite close, the effect is clearly detectable over a yearly period: less than one third among the agents whose education is lower than high school degree come to know professional forecasts over one year whereas nearly 40% of agents equipped with a college degree acquire the information. Equivalently, agents in the *lths* group are acquainted on average with the professional forecasts once in three years, whereas the ones with a degree typically need approximately nine months less.

Table 4 contrasts the distributions of the simulated and original data. The mean and the standard deviations are closely matched for any education level. Even more interestingly, the simulated data closely reproduce the minima, maxima and quartiles of the observed data. Differences are small and seldom exceed

[8] In the interpretation of the results, it has to be remembered that some of the individuals (on average, a fraction $\lambda^{edu} \cdot \mu^{edu}$) receive the signal o_t but still provide a stubbornly pessimistic answer. This implies that the fraction of individuals who receive the signal o_t *and really* incorporate it into the answer ans_{it} is approximately $\lambda^{edu} \cdot (1 - \mu^{edu})$, that is 0.059 for *lths*, 0.070 for *hs* and 0.083 for *col*.

Table 4. Summary statistics - simulated and original (1995Q1–2017Q2)

	Min	1st Qu	Median	Mean	3rd Qu	Max	St.Dev	Cor
lths - simulated	1.70	37.98	52.13	53.33	69.93	106.80	25.12	0.67
lths - original	4.97	35.84	50.40	53.34	64.90	114.00	24.79	
hs - simulated	−16.00	27.11	43.93	44.20	62.68	106.10	28.17	0.70
hs - original	−9.60	24.89	39.73	44.27	58.32	113.70	28.35	
col - simulated	−24.60	19.94	39.90	39.49	59.46	108.00	31.90	0.73
col - original	−23.47	15.77	32.70	39.09	54.32	109.50	32.39	

5 points (over a possible range of 400 points). The table also shows that the correlation between the real and the simulated indexes x_t and y_t^{edu} is about 70% and indeed turns out to be higher than the correlation between the index and the raw series of professional forecasts y_t^{edu} and o_t.

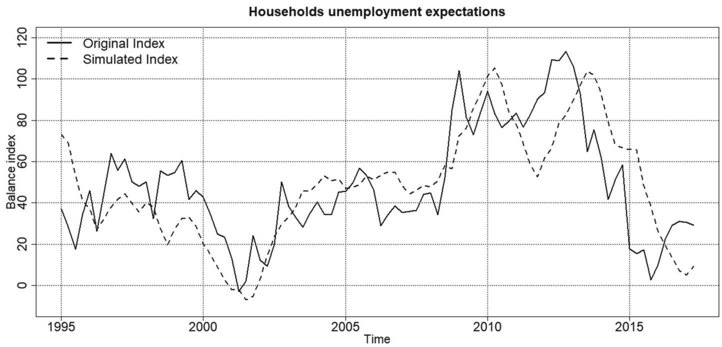


Fig. 3. Households unemployment expectations balance (1995Q1–2017Q2)

Figure 3 provides a visual comparison of the original and simulated index. Confirming the interpretation of Table 4, the fit is quite good especially in the first decade of the XXI century and, despite some misalignment around 2012, artificial data are reasonably close to their empirical target.

5 Conclusion

Our research question revolved around the observation that unemployment forecasts produced by surveyed "men in the street" are vastly different and, typically, stickier and gloomier, than the ones produced and disseminated by professionals. This is clearly evident contrasting the unemployment balance index provided by ISTAT with the data extracted from the Economic Outlook by OECD, that was used in this work to proxy the professional opinions of rational experts and respected institutions.

We developed an agent-based model where the respondents to the survey are typified by different education, affecting their willingness or ability to use the professional forecast as well as the frequency with which they stubbornly claim that "unemployment is likely to raise". The acquisition of the professional signal and the way it is processed (or "translated", to use the terminology of Sect. 2) are both stochastic events driven by different individual parameters. ABMs are well known to allow for the possibility to exploit the interactions of many boundedly rational agents who behave in a realistic way and are not required to be perfectly informed and able to analyze complex economic problems. As Conlisk [18] pointed out, rational choices can only be justified in some circumstances, say in the presence of a simple context, when good feedback can be obtained and there are plenty of opportunity to validate alternative decisions/estimates. The task of forecasting future unemployment rate is indeed demanding, nuanced and plagued with structural uncertainty and noise and, we believe, an ABM is a promising tool to reconcile the data with some plausible description of agents' behavior.

The model shows that scattered update of information together with the frequent recourse to over-pessimistic forecasts and oblivion are sufficient to accurately replicate the empirical data.

The results tell a story in which only a minority of agents acquire (i.e., is infected with) the public professional forecast and answer the survey accordingly. A much larger proportion of households, for reasons that are not investigated in this work but may be related, say, to extreme risk aversion or lack of trust in the official statistics, embrace the most pessimistic view. Most of the agents, yet, give an intermediate and lazy answer based on outdated information, whose effect can last with dubious effectiveness for several years. Observe that such agents may emotionally feel in a comfortable herd, as they belong to the majority of the population, see Baddeley [19].

One of the main novelty of our model is the introduction of education as a powerful driver of heterogeneity in behavior and the intuition that the schooling level is extremely relevant is a valuable insight of the model: more educated agents more often obtain and use the professional forecasts and less often resort to the cheap answer that unemployment will increase with no regard to the current economic situation. The relationship between education and "more rational" behavior turns out to be monotone and extremely significant in our agent-based setup.

It would be interesting to explore in future research the alternative assumption that there is peer-to-peer (local) information exchange (by contrast, in the present framework there is a centralized and unique source for the rational forecast). The NetLogo code would allow for similar computational generalizations that implement other strains of agents' bounded rationality.

References

1. Carroll, C.D., Hall, R.E., Zeldes, S.P.: The buffer-stock theory of saving: some macroeconomic evidence. Brookings Papers Econ. Act. **1992**(2), 61–156 (1992)
2. Hegselmann, R., Krause, U.: Opinion dynamics and bounded confidence models, analysis, and simulation. J. Artif. Soc. Soc. Simul. **5**(3) (2002)
3. Castellano, C., Fortunato, S., Loreto, V.: Statistical physics of social dynamics. Rev. Modern Phys. **81**(2), 591 (2009)
4. Hegselmann, R., König, S., Kurz, S., Niemann, C., Rambau, J.: Optimal opinion control: the campaign problem. J. Artif. Soc. Soc. Simul. **18**(3) (2015)
5. Carroll, C.D.: The epidemiology of macroeconomic expectations. In: Blume, L.E., Durlauf, S.N. (eds.) The Economy as an Evolving Complex System, III: Current Perspectives and Future Directions, pp. 5–29. Oxford University Press, Oxford (2006)
6. Rabin, M.: Psychology and economics. J. Econ. Lit. **36**(1), 11–46 (1998)
7. Camerer, C.F.: Progress in behavioral game theory. J. Econ. Perspect. **11**(4), 167–188 (1997)
8. Branch, W.A.: The theory of rationally heterogeneous expectations: evidence from survey data on inflation expectations. Econ. J. **114**(497), 592–621 (2004)
9. Axtell, R.L., Epstein, J.M.: Coordination in Transient Social Networks: An Agent-Based Computational Model of the Timing of Retirement, pp. 161–183. Brookings Institution Press, Washington DC (1999)
10. Moro, A., Pellizzari, P.: A computational model of labor market participation with health shocks and bounded rationality. Knowl. Inf. Syst. **54**(3), 617–631 (2018)
11. Lusardi, A., Mitchell, O.S.: Financial literacy around the world: an overview. J. Pension Econ. Financ. **10**(4), 497–508 (2011)
12. Souleles, N.S.: Expectations, heterogeneous forecast errors, and consumption: micro evidence from the Michigan consumer sentiment surveys. J. Money Credit Bank. **36**(1), 39–72 (2004)
13. Easaw, J., Golinelli, R., Malgarini, M.: What determines households inflation expectations? Theory and evidence from a household survey. Eur. Econ. Rev. **61**, 1–13 (2013)
14. Epstein, J.M.: Why model? J. Artif. Soc. Soc. Simul. **11**(4), 12 (2008)
15. Epstein, J.M.: Generative Social Science: Studies in Agent-Based Computational Modeling. Princeton University Press, Princeton (2006)
16. Capistrán, C., Timmermann, A.: Disagreement and biases in inflation expectations. J. Money Credit Bank. **41**(2–3), 365–396 (2009)
17. Wilensky, U.: NetLogo (1999). http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL
18. Conlisk, J.: Why bounded rationality. J. Econ. Lit. **34**, 669–700 (1996)
19. Baddeley, M.: Herding, social influence and economic decision-making: socio-psychological and neuroscientific analyses. Philos. Trans. Roy. Soc. Lond. B: Biol. Sci. **365**(1538), 281–290 (2010)

Towards Reducing Complexity of Multi-agent Simulations by Applying Model-Driven Techniques

Benjamin Hoffmann[1(✉)], Kevin Chalmers[2], Neil Urquhart[2], Thomas Farrenkopf[1], and Michael Guckert[1]

[1] KITE - Kompetenzzentrum für Informationstechnologie, Technische Hochschule Mittelhessen, Gießen, Germany
{benjamin.hoffmann,thomas.farrenkopf,michael.guckert}@mnd.thm.de
[2] School of Computing, Edinburgh Napier University, Edinburgh, Scotland
{k.chalmers,n.urquhart}@napier.ac.uk

Abstract. Creating multi-agent simulations is a challenging task often requiring programming skills at the professional software developer level. Model driven methods of software development are an appropriate tool for reducing the complexity of the development process of such simulations. The modeller is relieved from implementing time consuming programming details and can concentrate on the application itself. We present the domain specific language Athos with which network based traffic simulations can be created declaratively. The models are platform independent and executable code can be generated for two popular multi-agent platforms. We use a simple yet illustrative example to show how Athos can be applied.

Keywords: Domain-Specific Language · Model-driven development Traffic simulation

1 Introduction and Motivation

Agent-based simulations are an effective technique to model and analyse systems in which the overall behaviour is determined by the behaviour of its constituent autonomous entities [28]. Generally, Agent-Based Modelling (ABM) is a challenging task [5,25,31]. In comparison to more traditional modelling approaches like system dynamics, ABM has an increased level of cognitive complexity. Vendrov et al. argue that one reason for the higher complexity can be found in the programming languages used in current ABM frameworks [31]. These languages provide a low-level abstraction which makes it hard to create, understand and validate agent-based models. This is especially true for domain experts who are not professional programmers. Additionally, the complexity of ABM has increased due to grown computing power allowing larger models with more complex interactions [25]. At the same time, the domains analysed by ABM have

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 187–199, 2018.
https://doi.org/10.1007/978-3-319-94580-4_15

become more complex themselves and thus require more powerful models [1]. Development of agent-based simulations requires insight into two distinct bodies of knowledge [3]: the knowledge and experience of experts in the respective domain; and skills in software development.

It seems obvious that domain experts should work closely with software engineers to achieve high-quality agent-based simulations. In practice, the cooperation often leads to new problems like inconsistent terminologies [22] or different expectations regarding competences [29]. Therefore, scientific software is often designed and implemented by the same scientists who will eventually use it for their work. These scientists usually lack fundamental software engineering expertise [18]. North and Macal consider the development of interfaces that allow domain experts with little programming knowledge to create agent-based models as one key challenge in the field of ABM [23].

Having successfully implemented an agent-based model, verification and validation pose the next challenges [6]. While verification ensures that a model implementation is congruent with its conceptual specification, validation guarantees that the implementation accurately represents reality [17,33]. If implementation languages require an increased effort to understand a model implementation, they also make verification and validation more difficult. In addition, current approaches impede replication as they are often directly interwoven with their target platform [28].

The use of Domain-Specific Languages (DSLs) constitutes a promising approach towards the solution of the aforementioned problems (c.f., e.g. [21,23]). By production of intermediate models that bridge the gap between the researcher's abstract non-formal models and the final model implementation, DSLs present well-sized building blocks to create models from which simulations can automatically be generated. As the complexity of implementation details is hidden from language users, models become more comprehensible and focused. Increased abstraction also facilitates reproducibility of models and thus ensures high-quality simulation results in the scientific community.

In this paper, we present a DSL called **Athos** which supports researchers in the development of agent-based simulations. Athos allows declarative specifications for *traffic simulation* scenarios and thus relieves modellers from complex programming tasks. Major benefits of this approach are a clearer separation of concerns in the overall simulation model and enhanced support for domain experts in the set-up of agent-based simulations.

The remainder of this paper is organized as follows. Section 2 provides background on DSLs and agent-based simulations and discusses how these research fields have been brought together in related projects. Section 3 explains how Athos raises the abstraction level for agent-based models and thus alleviates the aforementioned problems. Section 4 presents a case study that applies Athos to a simple yet illustrative problem. Section 5 concludes this paper and shows some directions for future work.

2 Background and Related Work

A Domain Specific Language (DSL) is a programming language that is tailored towards a specific problem domain. Advantages of DSLs include decreased development time and reduced requirements of software development skills, allowing a domain expert to develop software for the given domain. When constructing models, a DSL allows a stronger focus on the model and domain issues rather than on lower-level programming issues [7,10]. Exploring complex problems through computer based simulation and modelling has been widely recognised as a useful means of increasing our understanding of real-world problems. A number of software frameworks for simulation and the associated analysis exist. These range from generalist frameworks such as SimStudio [30] to domain specific simulations such as MatSim [16] designed specifically for the transportation domain. A recent overview of the development and use of domain-specific modelling is given by Çetinkaya [4]. According to [16], such software models may be utilised via software packages aiming directly at the domain user. This often leads to models constrained to the functionality offered by the packages' user interface. If a model is incorporated within a DSL, the user may use the DSL to configure the model offering a far wider range of possibilities.

Multi-Agent Systems (MAS) are regarded as a useful means to construct simulations. Most simulations revolve around the interactions of a set of entities such as: people or vehicles [11]; stock market shares [24]); or consumers within a marketplace [9]. In such scenarios, the entities in the system being modelled can be represented by specific software agents. Ge and Polhill [11] use a MAS to investigate the actions of commuters and how their decisions are influenced by changes in the road network (specifically, the addition of new road links). In this case groups of commuters are represented by a software agent that takes decisions from the perspective of that individual.

Most software-based simulations are carried out for the benefit of specialists in other fields. This leads to a paradox. The non-computing specialist is unable to construct simulations using a framework such as the Java Agent Development Environment (JADE) [2] and instead uses an existing simulation package (e.g. MatSim [16]). Package-based simulation is restricted by the functionality provided by the package which may or may not be sufficient for the task being considered. If no package exists that can support the desired simulation, then a platform such as JADE or NetLogo [32] must be used by someone with the appropriate specialised skills. A DSL is a programming language that is tailored towards a specific domain (including Software Engineering [7] or MAS [12]). It requires less skill than a traditional programming language. This gives two distinct advantages: the language can be learned by a domain expert; and the process of model development is simplified.

Within transportation, a DSL has the potential to isolate the modeller from issues such as handling details of maps, coordinate systems or routing. This frees the modeller to concentrate on issues such as defining decision making and making explicit the problem constraints. This has two useful outcomes. Firstly, the development process may be accelerated. Secondly, the development may be undertaken by a domain specialist rather than a software engineer.

A contribution in that direction is made by Hassan et al. [14] who define a process to develop models for a given domain in the field of social sciences. The authors base their approach on the INGENIAS methodology [26] that comprises a meta-metamodel from which further metamodels can be derived. INGENIAS provides the necessary tools to develop a working graphic editor for the defined (meta-)models together with the necessary transformations to the intended target platform. The presented approach is designed so that it can potentially be applied to any agent-based modelling and simulation approach within the confines of social sciences. On the one hand, this makes the presented approach highly flexible and applicable for a wide range of problems. On the other hand, it requires users to develop a suitable metamodel for their problem as well as suitable transformations and cannot be applied in an out-of-box style.

GAMA [13] is another contribution that aims to facilitate the creation and simulation of complex agent-based models through employment of a platform-internal DSL. Models are described in GAML, a DSL that allows to define multiple layers within a simulation (e.g. one layer for the inside of buildings and another for the outside world buildings are located in), handling of geographical data, and definition of agents' attributes and behaviour. Though GAML is an agent-oriented DSL, it also allows to describe parts of a model by means of ordinary differential equations. In contrast to our approach, GAML is domain-specific in that it is agent-oriented, but it does not aim at a specific application domain. This makes it difficult to describe models in a pure or mostly declarative way but requires users to define agents' behaviour through *actions* and *reflexes* in a more procedural manner.

3 Athos – A DSL for Traffic Simulations

Athos is a DSL that seeks to support the development of MAS simulations. The language focuses on traffic scenarios that involve vehicles (agents) with individual behaviour (e.g. finding shortest routes). Thus, an Athos program involves a multitude of individual-level optimisation problems that each affect the state of the global system. Athos allows scenario definition in a declarative manner. This relieves users from complex programming tasks and enables them to focus on *what* to simulate instead of *how* to simulate it (c.f. [3,31]).

Figure 1 illustrates the main components of the language and its flow of information. The creation of a traffic simulation with Athos is based upon the development of a conceptual model which features aspects from the domain of traffic optimisation. These models allow to describe information on aspects that are relevant in this context, e.g. the capacity of certain roads in an area of interest or how certain types of vehicles congest certain roads. However, these models do not contain any information on computational details. Thus, from a computational view, they are created on the most abstract level and considered as Computationally Independent Models (CIM) [19].

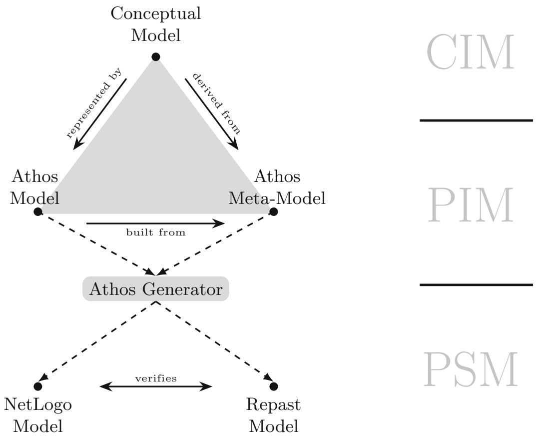


Fig. 1. Athos' modelling approach.

The language elements of Athos were derived from the CIM level in order to make them available to models which also consider aspects that are relevant for computer simulations. While these models are more specific than their CIM ancestors, they still do not allow any assumptions with regard to the implementation platform. Models on this level are said to be Platform Independent Models (PIM) [19,28]. Every program written in Athos constitutes such a PIM. Since the meta-model elements from which Athos programs are built are directly derived from the domain of traffic optimisation problems, each Athos PIM concisely represents the underlying CIM of the traffic domain.

The Athos generator finally processes models from the PIM level and transforms them into models for a specific simulation platform known as Platform Specific Models (PSM). In order to transform a PIM into a PSM, the generator has to add platform-specific details to the information drawn from the PIM. This is done by means of code templates which are created for every supported target platform. Currently, Athos features templates for two agent-based simulation platforms. The first is the NetLogo[1] platform, whose models follow a procedural paradigm. The other platform is the Repast Simphony[2] platform for which models are constructed in an object-oriented way. As is pointed out by Sansores and Pavón, the creation of models for different target platforms supports validation and verification efforts as both simulation implementations should present equivalent results [28].

The main tool used in the development of the language is the Xtext[3] language workbench (version 2.12). Next to the definition of the abstract and concrete syntax of the language, the workbench was also used to define transformations that constitute the language's dynamic semantics.

[1] https://ccl.northwestern.edu/netlogo/.
[2] https://repast.github.io/download.html.
[3] https://www.eclipse.org/Xtext/.

Athos is developed in an iterative and incremental manner. Although Athos is ultimately intended to support the development of various different optimisation-related traffic scenarios, it focuses on one part of the problem domain at a time. The language's first major development iteration focused on scenarios in which agents seek to get from a given starting point in a road network to a predefined destination. For this, they seek to find the route that requires the least amount of time. Since in the underlying network the amount of time it takes to travel a given road is dependent on the current traffic situation, each agent is confronted with a dynamic optimisation problem.

Traffic-related problems necessitate the definition of some kind of network. Athos allows specification of road networks that consist of nodes and edges in a straightforward manner. In order to populate the network with agents, nodes can be defined as sources from which new agents originate. Users can specify time-based patterns in which agents are created. It is also possible to model distribution functions. These functions control how source nodes spawn agents with different properties into the system.

Agents differ in two major properties. Firstly, agents can be assigned different routing modes that determine how they move inside the network. Most of the time, modellers will assign an a priori destination to agents. However, in order to generate background noise or to simulate public transport routes, it is also possible to define agents that circle or shuttle along a pre-defined sequence of nodes in the network. Secondly, agents can be assigned a congestion factor. The congestion factor is a value that determines to what extend the respective agent will congest, i.e. slow down, traffic on the road it travels on. This way, agents can represent different types of vehicles. An agent with a high congestion factor could represent a bus or a tractor whereas an agent with a low congestion factor could represent a fast car or a motorcycle.

As was already stated, agents that head towards a pre-defined destination try to get there in the least possible amount of time. Vehicle agents calculate the fastest path from their current location to their intended destination node by means of Dijkstra's algorithm [8]. They do so every time they enter a node of the network. Whenever agents recalculate the fastest path, they consider the current traffic situation in the network. For this, roads must be assigned a numerical value that represents the amount of time cars need to travel them. To this end, Athos allows the definition of cost functions as ordinary mathematical expressions. Within these expressions, various properties like the length of the road or the accumulated congestion factor of all vehicles can be used. The value of the accumulated congestion factor for a given road depends on both the number and the types of agents that are on this road at a given point in time.

The described language features allow for the definition of traffic-dependent travel durations. The higher the accumulated congestion factors of all vehicles on a given road, the longer it takes these vehicles to get to the next road. This, in turn, increases the time window in which other vehicles can further increase the accumulated congestion factor on this road. This way, the language allows for the definition of scenarios where increased numbers of agents congest certain

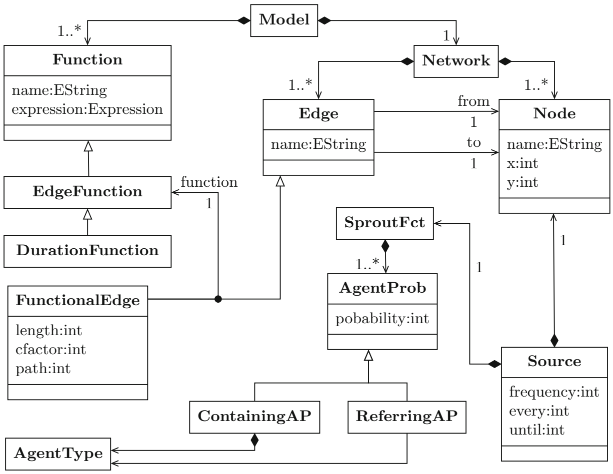


Fig. 2. Simplified excerpt of Athos' meta-model

roads so that traffic may ultimately grind to a halt. By default, all simulation scenarios track the total amount of time cars have spent in the system.

Figure 2 shows an excerpt of the language's meta-model. Any program written in Athos is a `Model`. A program features several types of `Functions`. Some of these are used to influence the way an agent travels a given edge. Such functions are called `EdgeFunctions`. This is because they are always associated with an arbitrary number of `Edges`. Agents that travel an edge have to follow the rules implied by the function associated with this edge. As will be shown shortly, there is also another type of function that is used by (or associated with) agents. `EdgeFunctions` are further specialised to `DurationFunctions`. They feature an expression that determines the amount of time it takes to cross a certain edge. Later versions will feature additional specialisations of the `EdgeFunction` class, e.g. speed-dependent functions whose expressions determine the speed by which agents drive on a given edge.

Each program must also feature a `Network` which consists of `Nodes` and `Edges`. Nodes and edges must be identifiable by a unique `name`. Additionally, nodes need coordinates to locate them in the plane. Edges need both a node `from` which they emerge and a node `to` which they lead. Nodes can assume the role of a `Source` that sprouts agents into the network. Source nodes are always associated with a sprout function (`SproutFct`). A sprout function contains a (non-empty) set of agent probabilities (`AgentProb`). Both specialisations, i.e. `ContainingAP` and `ReferringAP`, have a reference to an `AgentType` and an integer that represents the probability with which the referred type of agent is created. While a `ContainingAP` allows to define an anonymous agent type inside a sprout function, `ReferringAP` objects refer to a named agent type defined in an Athos program.

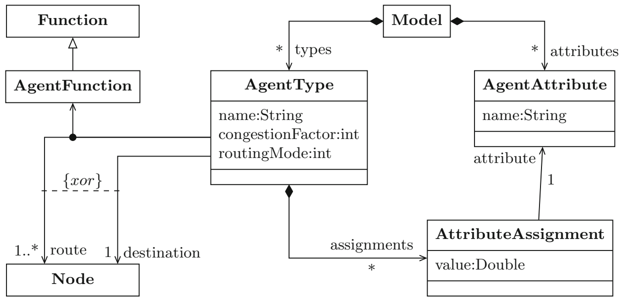


Fig. 3. Athos' `AgentType`-related meta-model elements.

Figure 3 illustrates how Athos allows for the definition of agent types together with type specific attributes and optimisation functions. An `AgentType` can either be named or anonymous depending on whether its `name` attribute was set. Each `AgentType` can be assigned an arbitrary number of attributes defined in the model (e.g. fuelConsumption). The connection between the type and the attribute is established by means of an `AttributeAssignment` which also assigns a value for the attribute. For this reason, all agents of the same type have the same value for a given attribute. As was already mentioned, different `AgentTypes` can be assigned different `congestionFactors` and `routingModes`. The attributes of an agent can be used inside an `AgentFunction`. An `AgentFunction` can be assigned to an `AgentType` so that all agents of this type will calculate their routes in a way that minimises the associated function.

4 Example

This case study demonstrates how Athos can be applied to a simple yet illustrative problem. We first consider a network of roads with a single source and a single target of traffic. Agents (i.e. cars) are created in the source node and use a shortest route to the target while travel time depends on congestion effects on the roads represented by the edges of the network.

Given such a traffic aware network with vehicles travelling from defined sources to individually defined targets, analysing the overall behaviour of this system soon becomes a complex task. While static instances (i.e. networks with static travel times not affected by congestion) of the problem can be solved with analytical methods, agent based simulations are an appropriate tool for dynamic versions of this problem. However, even simple networks quickly lead to complex programs in popular, rather easy to use agent based programming environments. In our case study we consider a network of eight nodes and nine edges. This network corresponds to Samuelson's example [27] in which the Braess paradox and the effect of modified cost functions determining the choice of the routes is examined.

```
1 model Model1 world xmax 60 xmin 0 ymax 60 ymin 0
2 functions // functions used in the model
3 durationFunction traveltime  length + cfactor * accCongestionFactor
4 network
5 nodes // nodes of the network
6   node a (2.0, 15.0)
7   node b (10.0, 15.0)
8   node c (20.0, 15.0)
9   node d0 (30.0, 15.0)
10   node t1 (8.0,25.0)
11   node t2 (15.0, 25.0)
12   node d1 (15.0, 5.0)
13   node d2 (23.0, 5.0)
14 edges // roads that connect the nodes
15   edge ab from a to b length 2.0 cfactor 0.02  function traveltime
16   edge cd from c to d0 length 2.0 cfactor 0.02 function traveltime
17   edge at1 from a to t1 length 10.0 cfactor 0.002 path "ac" function traveltime
18   edge t1t2 from t1 to t2 length 10.0 cfactor 0.002 path "ac"  function traveltime
19   edge t2c from t2 to c length 5.0 cfactor 0.002 path "ac"  function traveltime
20   edge bd1 from b to d1 length 10.0 cfactor 0.002 path "bd"  function traveltime
21   edge d1d2 from d1 to d2 length 10.0 cfactor 0.002 path "bd" function traveltime
22   edge d2d from d2 to d0 length 5.0 cfactor 0.002 path "bd"  function traveltime
23   edge bc from b to c length 1.0 cfactor 0.03 baseSpeed 1.0 function traveltime
24 sources
25    a sprouts (congestionFactor 100.0 destination d0 ) frequency 25.0 every 1 until 3
```

Individual travel time is calculated by means of the duration function *traveltime* defined in line 3. This function accumulates the product of *cfactor* and *congestionFactor* over all individuals using the edge at the very moment and adds it to the length of the edge. Note that *congestionFactor* is an attribute of the travelling agent that determines how strong this individual agent will add to congestion while *cfactor* is an attribute of the edge and defines the (linear) congestion effect on this edge.

This PIM can either be generated to be run as a NetLogo or a Repast Symphony simulation in their respective environment. While the first offers the easy to use NetLogo system, the latter is created to be run in the scalable Java-based Repast Symphony system. Note that this model is also computation independent as we have argued in the previous chapters: we just define the problem without details of how the solution is to be computed leaving that as a task of the language and the architecture.

Comparing the model above with directly implemented simulations in both of the target platforms shows that the model is highly self-documenting. The code generated for either of the two platforms is difficult to understand even though the underlying templates follow best practice approaches for the platforms. Probably one of the most basic measures for software complexity is simply counting the number of statements or just lines of code. While the Athos model consists of 25 lines, the generated NetLogo source contains 572 lines and the Repast Symphony program 1531 lines in 8 classes. Note that according to the principles of model driven software development, the generated code is well structured and follows best practice guidelines and does not contain unnecessary statements.

We will discuss some exemplary language features of Athos by extending the model from above. For example, different types of agents with individual behaviour in how they determine their preferred route from source to destination can be defined. This can be done very intuitively by just defining an additional type of agent *ecoDriver* to the default type and an alternative function *ecoDriverFunction* to compute travel costs relevant to this type of agent. *ecoDriver* agents possess the attribute *fuelConsumption* available to the newly defined function.

```
...
agentAttributes fuelConsumption
agentTypes
agentType ecoDriver congestionFactor 50.0 attr fuelConsumption = 10.0 destination d_ optimises ecoDriverFunction
functions // functions used in the model
durationFunction traveltime  length + cfactor * accCongestionFactor
agentFunction ecoDriverFunction length * fuelConsumption
...
```

Furthermore, we can define two sources of traffic (a and b) both being the origin of agents heading towards the same destination d. This is achieved by replacing the last line of the model code by:

```
b sprouts (congestionFactor 100.0 destination d0 ) frequency 50.0 every 1 until 1
a sprouts (ecoDriver probability 95) , (congestionFactor 100.0 destination d0 probability 5) frequency 50.0 every 1 until 1
```

This increases the number of lines of code in the Athos model to 30 and to 587 in the generated NetLogo source.

Besides the number of lines of code as an obvious indicator for complexity, a closer inspection of the implementations shows how the code in both platforms contains a significant amount of instructions that can hardly be mapped to the simulation directly but are necessary implementation details a domain expert would not want to bother with. Following Crooks et. al., by applying the rule of parsimony (aka Occam's razor) we again get a strong argument for the quality of the Athos approach (see [6]).

5 Conclusion and Future Work

This paper argues that a model driven approach for MAS in a given domain (e.g. traffic as in our case study) can significantly reduce the complexity of the code that is visible to the domain expert. The domain expert creates platform independent models that are transformed into executable simulations. In this paper, we have shown how Athos allows the creation of PIMs from computationally independent conceptual models in order to define scenarios in which agents solve optimisation problems. In order to provide a deeper understanding of the language, we have discussed Athos' meta-model elements and how they are related. We have also shown that Athos models describe the underlying problem in a concise and declarative way. While the optimisation problems solved by the agents are rather simple on the individual level, they yield a complex system behaviour as all agents mutually affect their travel times in the network.

Athos was developed with an iterative approach and will continuously be extended in that way. Later versions of the language will allow to define agents that receive information on the current state of the network in a temporally deferred manner. This will allow experiments on the importance of timeliness of information in traffic networks. We will also introduce features that give fine-granular control on what variables are to be tracked and visualised. Currently, we are working on agents that can find optimal tours for a given set of nodes and congestion dependent travel times in the network, i.e. these agents are facing instantiations of a dynamic Travelling Sales Problem.

Even though the number of lines of code required to define optimisation-related traffic simulations can be dramatically reduced, there is a need to quantitatively evaluate how this affects the cognitive complexity of the models. For this, Athos will have to be evaluated in an objective way by appropriate tests. In this context, we will also empirically evaluate qualitative DSL aspects like usability, productivity, learnability and reliability that are crucial for a successful DSL [15]. We consider such a quantitative evaluation of these aspects to be of utmost importance even though it is often neglected by language developers [20].

However, literature still offers little guidance on how DSLs can be systematically evaluated [20]. This might be because this part of DSL development is still at a rather immature stage [5]. For this reason, scientists should view it as an important and interesting and rather unexplored area of further research.

References

1. Bazzan, A.L.C., Klügl, F.: A review on agent-based technology for traffic and transportation. Knowl. Eng. Rev. **29**(03), 375–403 (2014)
2. Bellifemine, F., Bergenti, F., Caire, G., Poggi, A.: Jade—a Java agent development framework. In: Bordini, R.H., Dastani, M., Dix, J., El Fallah, S.A. (eds.) Multi-Agent Programming. Multiagent Systems, Artificial Societies, and Simulated Organizations, vol. 15, pp. 125–147. Springer, Boston (2005). https://doi.org/10.1007/0-387-26350-0_5
3. Borenstein, D.B.: Nanoverse: a constraints-based declarative framework for rapid agent-based modeling. In: Yilmaz, L. (ed.) Proceedings of the 2015 Winter Simulation Conference, pp. 206–217. IEEE, Piscataway (2015)
4. Çetinkaya, D.: A model driven approach to web-based traffic simulation. In: A Model Driven Approach to Web-Based Traffic Simulation (2016)
5. Challenger, M., Kardas, G., Tekinerdogan, B.: A systematic approach to evaluating domain-specific modeling language environments for multi-agent systems. Softw. Q. J. 1–41 (2015)
6. Crooks, A., Castle, C., Batty, M.: Key challenges in agent-based modelling for geo-spatial simulation. Comput. Environ. Urban Syst. **32**(6), 417–430 (2008)
7. Van Deursen, A., Klint, P.: Little languages: little maintenance? J. Softw. Maint.: Res. Pract. **10**(2), 75–92 (1998)
8. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numer. Math. **1**(1), 269–271 (1959)
9. Farrenkopf, T., Guckert, M., Urquhart, N., Wells, S.: Ontology based business simulations. J. Artif. Soc. Soc. Simul. **19**(4), 14 (2016)
10. Fowler, M.: Domain Specific Languages, vol. 1. Addison-Wesley Professional, Boston (2010)
11. Ge, J., Polhill, G.: Exploring the combined effect of factors influencing commuting patterns and CO2 emissions in Aberdeen using an agent-based model. J. Artif. Soc. Soc. Simul. **19**(3) (2016)
12. Grey, R.: Agent Tcl: a transportable agent system. In: Proceedings of the CIKM Workshop on Intelligent Information Agents, Fourth International Conference on Information and Knowledge Management (CIKM 1995) (1995)
13. Grignard, A., Taillandier, P., Gaudou, B., Vo, D.A., Huynh, N.Q., Drogoul, A.: Gama 1.6: advancing the art of complex agent-based modeling and simulation. In International Conference on Principles and Practice of Multi-agent Systems, pp. 117–131 (2013)

14. Hassan, S., Fuentes-Fernández, R., Galán, J.M., López-Paredes, A., Pavón, J.: Reducing the modeling gap: on the use of metamodels in agent-based simulation. In: 6th Conference of the European Social Simulation Association (ESSA 2009), pp. 1–13 (2009)
15. Hermans, F., Pinzger, M., van Deursen, A.: Domain-specific languages in practice: a user study on the success factors. In: Schürr, A., Selic, B. (eds.) MODELS 2009. LNCS, vol. 5795, pp. 423–437. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04425-0_33
16. Horni, A., Nagel, K., Axhausen, K.W.: The Multi-Agent Transport Simulation MATSim. Ubiquity Press, London (2016). https://doi.org/10.5334/baw
17. North, M.J., Macal, C.M.: Agents up close. In: North, M.J., Macal, C.M. (eds.) Managing Business Complexity, pp. 24–44. Oxford University Press, Oxford (2007)
18. Joppa, L.N., McInerny, G., Harper, R., Salido, L., Takeda, K., O'hara, K., Gavaghan, D., Emmott, S.: Troubling trends in scientific software use. Science **340**(6134), 814–815 (2013)
19. Kardoš, M., Drozdová, M.: Analytical method of CIM to PIM transformation in model driven architecture (MDA). J. Inf. Org. Sci. **34**(1), 89–99 (2010)
20. Kosar, T., Bohra, S., Mernik, M.: Domain-specific languages: a systematic mapping study. Inf. Softw. Technol. **71**, 77–91 (2016)
21. Kosar, T., Mernik, M., Carver, J.C.: Program comprehension of domain-specific and general-purpose languages: comparison using a family of experiments. Emp. Softw. Eng. **17**(3), 276–304 (2012)
22. Lu, R., Jin, Z.: Domain Modeling-based Software Engineering: A Formal Approach, vol. 8, p. 123. Springer, New York (2000). https://doi.org/10.1007/978-1-4615-4487-6
23. North, M.J., Macal, C.M.: Agent based modeling and computer languages. In: Meyers, R.A. (ed.) Encyclopedia of Complexity and Systems Science, pp. 131–148. Springer, New York (2009). https://doi.org/10.1007/978-0-387-30440-3_8
24. Palmer, R.G., Arthur, W.B., Holland, J.H., LeBaron, B.: An artificial stock market. Artif. Life Robot. **3**(1), 27–31 (1999)
25. Parry, H.R.: Agent based modeling, large scale simulations. In: Meyers, R.A. (ed.) Encyclopedia of Complexity and Systems Science, pp. 148–160. Springer, New York (2009). https://doi.org/10.1007/978-0-387-30440-3_9
26. Pavon, J., Gomez-Sanz, J.J., Fuentes, R.: The INGENIAS methodology and tools. In: Henderson-Sellers, B., Giorgini, P. (eds.) Agent-Oriented Methodologies, pp. 236–276. IGI Global, Hershey (2005)
27. Samuelson, P.A.: Tragedy of the open road: avoiding paradox by use of regulated public utilities that charge corrected knightian tolls. J. Int. Comp. Econ. **1**(1), 3–12 (1992)
28. Sansores, C., Pavón, J.: Agent-based simulation replication: a model driven architecture approach. In: Gelbukh, A., de Albornoz, Á., Terashima-Marín, H. (eds.) MICAI 2005. LNCS (LNAI), vol. 3789, pp. 244–253. Springer, Heidelberg (2005). https://doi.org/10.1007/11579427_25
29. Segal, J.: Software development cultures and cooperation problems: a field study of the early stages of development of software for a scientific community. Comput. Support. Coop. Work (CSCW) **18**(5), 581 (2009)
30. Touraille, L., Traoré, M.K., Hill, D.R.C., A model-driven software environment for modeling, simulation and analysis of complex systems. In: Proceedings of the 2011 Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium, TMS-DEVS 2011, pp. 229–237. Society for Computer Simulation International, San Diego (2011)

31. Vendrov, I., Dutchyn, C., Osgood, N.D.: Frabjous: a declarative domain-specific language for agent-based modeling. In: Kennedy, W.G., Agarwal, N., Yang, S.J. (eds.) SBP 2014. LNCS, vol. 8393, pp. 385–392. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-05579-4_47
32. Wilensky, U.: Netlogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL (1999). http://ccl.northwestern.edu/netlogo/
33. Xiang, X., Kennedy, R., Madey, G., Cabaniss, S.: Verification and validation of agent-based scientific simulation models. In: Agent-Directed Simulation Conference, pp. 47–55 (2005)

Check for updates

Environment for Identification of Significant Subjects on Information Portals

Jarosław Koźlak[1(✉)], Małgorzata Żabińska[1], and Yves Demazeau[2]

[1] AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Kraków, Poland
{kozlak,zabinska}@agh.edu.pl

[2] CNRS – LIG, Grenoble, France
Yves.Demazeau@imag.fr

Abstract. A large amount of information available on the Internet makes it difficult to identify the most significant or interesting (for a given user) facts and relations between them. It especially concerns the content of information portals, e.g. provided by popular newspapers and magazines.

A pilot version of the software environment for aggregating such information and extracting interesting topics and relationships between them for the user is proposed. Techniques of multi-agent approach, data mining and complex network analysis are used. Evaluation is done by analysing a set of press notes made available by the world's most popular magazines and contents provided by Polish information portals.

1 Introduction

There is a huge amount of information available on the Internet which makes that it is hard to distinguish the most important facts and relations among them. It is possible to undertake an attempt to transform the information such that the most essential or interesting facts or dependencies may become visible by application of artificial intelligence techniques, multi-agent systems and data mining approaches. It enables to overcome complexity and distinguish contents directly related to subjects searched by us.

In our work we present the information and relations between these pieces of information using graph representation. The multi-agent approach is used to identify relations between pieces of information.

Knowledge is presented in a form of complex network where the interesting notions play the role of nodes, whereas relations between them are described by vertices.

Our goal is to create an environment which automatically analyses information channels, especially these related to press or information portals and provides following functions:

© Springer International Publishing AG, part of Springer Nature 2018

Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 200–211, 2018.
https://doi.org/10.1007/978-3-319-94580-4_16

– finds the most interesting contents from the point of view of a user,
– categorises contents available on the Internet.

We use techniques of agent approach and social network analysis [5].

The basis of the system development is an application of multi-agent approach. It enables modelling current relations between notions of information as well as their variability in time on a very local and detailed level. The multi-agent systems representing information available in the examined sources aggregates occurrences of particular notions contained in different sources of information together with additional information (such as e.g. in which sources in given times the notion was contained). Methods of complex networks allows to express total characteristics of the analysed information state, both in the whole considered time period an in some short time intervals.

In the paper we concentrate on analysis of specific sort of information as references to different countries mentioned in press notes are. The press notes come from sets of press notes about international politics collected from RSS feeds. They origin from several dozen of popular world-wide journals collected by the Geomedia project (as described in [9]). The aim is to better understand the specificity of individual notions, dependencies between them as well as influence of time upon the dependencies.

2 Related Work

2.1 Multi-agent Systems and Organisation of Society

Descriptions of interactions between agents are frequent elements of the multi-agent system models, they cause emerging of organisations in societies of agents [2].

Agents in societies [6] influence one another, depending on the frequency of their interactions they may have values representing their strengths assigned, which may change while time passes. The model of the multi-agent system described in this paper is organised as a society of agents connected in a networks using aggregation agents as mediators. The aim of our system is the analysis of information acquired dynamically from information portals and development of representation of events being causes of their taking place, in following section we will focus on different ways of information gathering.

An important class of multi-agent systems is this one which may describe their organisations in given times as networks. This allows to use different graph measures [13] for a description of agent position in society and its role in information flow.

In [4,5,7] we described different kinds of simulators modelling behaviour of social systems representing social media sites or blog portals thus systems where information analysis is a significant problem to handle.

2.2 Information Features and Organisation

Currently, users are overwhelmed by an excess of information that needs to be filtered and interpreted. Due to the limited time and resources available to individual users, it is important to automate these processes properly. One can proceed here in various ways, for example consider:

– aggregation of information – combining the related information from the point of view of accepted criteria in such a way as to limit their complexity [9],
– searching for information closely related to many other different information [8,9],
– identifying causal relationships between information [10],
– search for unusual information that differs from others with similar characteristics.

Due to the complexity of issues associated with information acquisition, integration and understanding, the use of multi-agent approach which offers autonomous and intelligent heterogeneous components of the system, combined with advanced methods of machine learning, data mining and complex network analysis can be a useful research direction to handle such problems.

Activities related to proper categorisation of information and their aggregation with a given level of detail are very important tasks and have been the subject of research. The multiagent approach is suitable for such interpretation of information [8,9].

In [9], the Kullback-Leibler Divergence measure was used to determine the information lost during the aggregation process. Groups of information (groups of agents) have been defined with a given number in order to carry out such aggregation in accordance with the assumed additional criteria [8].

These works were carried using press notes data obtained from Geomedia project [3], which are also used in the system and analyses presented in this paper.

This article focuses on the construction of a multi-agent system that can be used to solve a wide range of problems defined above, and focuses on selected test problems on identifying similarities in different sources of information and identifying the most significant information most closely related to other information.

In a situation where information is organised in the form of a network with a list of connected nodes, you can use social network analysis techniques or complex networks that provide a series of measures to describe different aspects of the vertex's significance in the graph to extract the most relevant information. In particular, the number and meaning of information indicating the information may be taken into account [11].

Numerous works were done, aimed at identification of similarities between information presented in information sources. If the content of information sources is represented as a text, a vector space model may be used [12].

3 Model Description

3.1 Theoretical Metamodel

The aim of the built multi-agent system is to maintain the current relations between notions in time. The relations may undergo dynamic changes. The notions describe the state of the observed information systems and the relations between them, among others – the their strength. We assumed that to describe relations of such information and dependencies, the agent model can be useful where agents represent single notions, complex relations between groups of notions and monitor the external information systems to update changes originated as a result of different (often undefined) events.

The single periods when the information is aggregated are distinguished. For each period the relations between the notions are specified. The notions are managed by agents assigned to them. The goal of the agents is to make the results of inference available. The results contain the description of location of the given notion in a graph, its variability, characteristic features and relations with the environment (adjacent notions in a graph). We can additionally distinct different types of relations and it is possible to calculate various types of neighbourhood for these relations.

This way we can define a state of the system at given time t:

$$SS(t) = (A(t), I(t), f, S(t), E(t), P(t)) \tag{1}$$

where:

- set of *Notion Agents* in the system, in time t – $A(t)$,
- a set of *Aggregation Agents* (which binds notions occurred in the same documents) – $I(t)$, neighbourhood relations are represented by single agents who are generated when needed and make connections with appropriate agents–notions; neighbourhood agents can connect with one or more notions,
- function assigning to *Aggregation Agents* their strengths (resulted from the number of common occurrence in documents) – f,
- set of observable sources, which are monitored by the system and notions occur in them – S,
- set of events $E(t)$ defining external events which influence the relations between notions, characterised by sets of notions associated with them,
- set of event patterns $P(t)$ describing the possible schemes of frequent events taking place.

Such defined states of the system may be subsequently submitted to analysis in different ways to define current states of notions and relations represented by agents. Particularly it is possible to perform analysis of the originated graphs created by notions and relations between them in single time spans, calculation of characterise measures and examination of their changeability. The threshold of acceptance of the change of the frequency of notion is set during the system configuration. If this threshold is exceeded it is necessary to use special agents

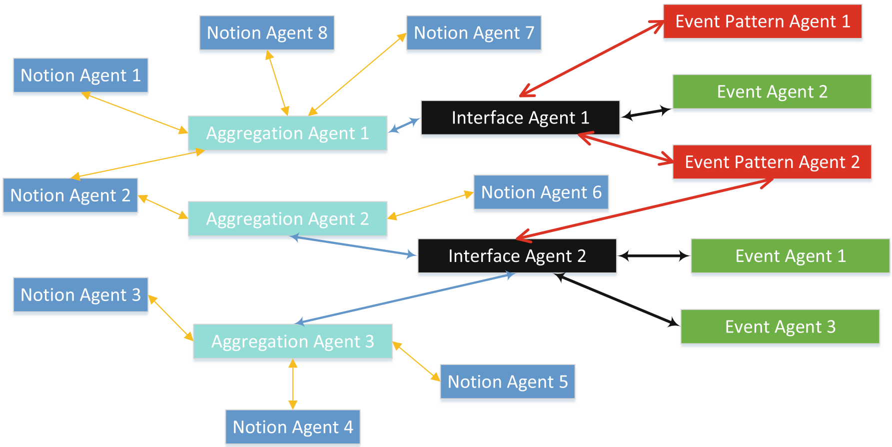


Fig. 1. Agent-based system model

(*Event Agents* and Event Pattern Agents) to explain reasons why the state of the information on considered information sources changed so significantly.

Figure 1 shows types of agents in the system and between which types of agents interactions my occur. Particular types of agents are described in the next section.

3.2 Agents

Notion Agents. Each considered piece of information serves as an agent called *Notion Agent*, which is defined as follows:

$$A_i = (N_i, F) \tag{2}$$

where:

- N_i – symbol of notion,
- F – set of frequencies of notion occurrences which elements take form of $f(N_i, s)$ or $f(N_i, t, s)$ representing occurrences currently and in the past in times t for different sources $s \in S$.

Actions taken by *Notion Agents*:

- create or destroy an Aggregation Agent,
- compute differences between groups in the subsequent time period.

Aggregation Agents. *Aggregation Agents* are responsible for making connections between agents, who often occurred together in press notes. They enable connection of notions which appear frequently together in information sources and are strongly connected. *Aggregation Agent* is permanently assigned to a group of *Notion Agents* and stores information on frequency of occurrence of relations between these notions.

Interface Agents. *Interface Agents* enable flexible observations of changes of relations and their frequencies between observed notions. They allow to monitor different characteristics of the system and state of notions, especially their frequencies (in total and in given information sources), frequent occurrences in the information sources of groups of notions, correlations among notions in different information sources and graph measures describing relations between notions.

Owing to this collected information they may create appropriate *Event Pattern Agents* which represent frequently appearing disturbances of intensities in time (e.g. associated with periodical conflicts between groups of countries) and *Event Agents* which represent disturbances which cannot be explained by periodical occurrence of increase of groups of notions.

Event Agents. These agents are used to model events that represent the changes of a state of the system, taking into consideration variability of frequency of relations between notions and existing event patterns definitions. These changes are strong enough that cannot be explained by statistical changes of content of information sources in given time period and by frequent pattern of events (represented by *Event Pattern Agents*. They are generated by *Interface Agents*.

Event Pattern Agents. These agents represent schemes of possible events which frequently appear, to easier identify their influence on the state of the system. When we consider the press notes concerning international politics they are associated with sets of countries which are often mentioned together because of belonging to the same organisations, close cooperation, common participation in wars, etc.

4 Testbed Environment

4.1 System Architecture

The developed pilot system contains following main modules:

- *Information Gathering* – gather new press notes from the monitored sources;
- *Tagger* – tool used to classify the information, tags press notes with keywords describing their content;
- *multi-agent system for Information Management* – represents the aggregated information and relations obtained from the database. It is based on the model presented in Sect. 3 and it used JADE platform [1];
- *Graph Controller* – responsible for generation of graphs representing the stored information and relations between considered notions;
- *Graph Handler* – responsible for visualisation of graphs; item Reporting Module – responsible for calculation of different graph measure to characterise the different aspects of importance of the considered notions in the aggregated information;

- *Interface* – responsible for the generation of queries to the database and obtaining the basis statistics about the information from all considered information sources;
- *Database* – stores gathered and tagged press notes (Fig. 2).

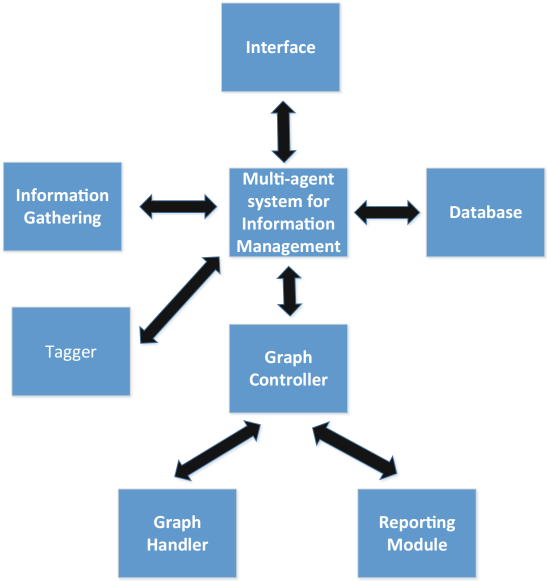


Fig. 2. System architecture

4.2 Geomedia Data

Data from 36 RSS feeds of popular world magazines on international events and collected during 18 months (1st January 2014 to 30th June 2015) coming from the GEOMEDIA project were selected as the test data.

The RSS feeds concerning world politics following popular journals (in parenthesis language of the notes and country of the origin of the journal) were analysed: (1) The Australian (en, AUS), (2) Herald Sun (en, AUS), (3) The Star (en, CAN), (4) China Daily (en, CHN), (5) South China Morning Post (en, CHN), (6) Daily Telegraph (en, GBR), (7) The Guardian (en, GBR), (8) Hindustan Times (en, IND), (9) The Times of India (en, IND), (10) Japan Times (en, JPN), (11) Times of Malta (en, MLT), (12) The Star (Malaisie) (en, MYS), (13) This Day (en, NGA), (14) New Zealand Herald (en, NZL), (15) The News International (en, PAK), (16) Today (en, SGP), (17) The New York Times (en, USA), (18) Washington Post (en, USA), (19) Chronicle (en, ZWE), (20) La Nacion (es, ARG), (21) La Razon (es, BOL), (22) La Patria (es, BOL), (23) El Mercurio (es, CHL), (24) La Tercera (es, CHL), (25) El Periodico de Catalunya (es, ESP), (26) El Pais (es, ESP), (27) La Jordana (Mex) (es, MEX), (28) El Universal (es, MEX), (29) El Universal (VEN) (es, VEN), (30) Derniere Heure (fr, BEL), (31) Le soir (fr, BEL), (32) Le Journal de Montreal (fr, CAN), (33) El Watan (fr, DZA), (34) L'Expression (fr, DZA), (35) Le Parisien (fr, FRA) and (36) Le Monde (fr, FRA).

5 Results

We assume that the feed coming from each journal is an independent monitored information source, which changes dynamically with the appearance of new press notes. For each information source we have separated 18 consecutive states containing notes collected within 1 month. We have analyse the following concepts in the sources of information: countries mentioned in press notes or identified frequent terms. In this article we focus on the analysis of information constituted by countries mentioned in the notes. The following analyses are presented in the section:

- the overall aggregate amount of information in each information source collected in each data source – number of times countries were mentioned in different press notes (Table 1),
- similarities between the topics discussed in different information sources determined by the calculated cosine measure of vectors, due to the limited space, some magazines have been omitted here (Fig. 3),
- aggregated information appearing in information sources concerning two and three most frequently mentioned groups of countries in press notes (Fig. 4),
- dynamic change the set of countries published by two selected information sources (Figs. 5 and 6),
- calculation of values of basic measures of complex networks for the most-mentioned countries, which determine their location in the generated information network (Table 2).

In Table 1 there are total frequencies of mentioning of the given countries in the press notes coming from different magazines in the considered dataset. One can notice that the countries which occur the most often are: USA, Ukraine, Russia, Chine, UK and France. Some journal such as *The Guardian*, *New Zealand Harald* and *Today* refer to quite different countries.

In Fig. 3 we have presented the cosine similarity measures describing how similar the content provided by given journals is. For the sake of transparency, only 10 journals that had the highest number of references to countries were selected and they were compared to three journals: *The China Daily*, *The Guardian* and *Le Monde*. We can notice significant differences in subjects presented by considered journals.

In Fig. 4 we have presented countries which are most frequently referenced together with other countries in groups consisting of 2 or 3 elements. The most numerous groups containing 2 countries are: RUS-UKR, ISR-PSE, RUS-USA, CHN-USA, IRQ-USA; while containing 3 countries are as follows: RUS-UKR-USA, DEU-RUS-UKR, FRA-RUS-UKR, DEU-FRA-UKR, DEU-FRA-RUS and ISR-PSE-USA.

Table 1. Total number of notes that mention given countries for each considered journal

Journal	USA	UKR	RUS	CHN	GBR	FRA	ISR	SYR	IRQ	PSE	IRN	AUS	DEU	IND	NGA	GRC	EGY	JPN	TUR	VAT
(1)	946	369	398	281	268	149	208	177	187	155	113	76	52	108	59	129	67	92	75	93
(2)	161	26	73	128	53	34	9	24	110	6	21	2604	23	46	1	8	19	51	7	13
(3)	850	595	468	201	251	266	328	252	208	261	113	124	107	253	222	62	197	90	124	144
(4)	2669	672	893	4796	446	317	217	191	176	199	224	357	243	400	87	122	154	1138	103	8
(5)	2849	751	822	581	754	447	270	269	244	218	182	226	175	39	124	118	162	51	130	110
(6)	4107	1506	1492	984	1758	1308	842	751	624	663	347	698	482	468	227	158	249	369	249	339
(7)	8069	2057	2310	1407	3313	1322	1277	1181	1011	1020	890	2830	737	973	370	1009	591	550	498	446
(8)	1406	256	255	495	414	225	229	227	349	231	112	151	69	504	162	41	100	176	76	66
(9)	2021	500	474	720	612	244	296	400	388	240	302	124	115	283	200	64	224	224	130	155
(10)	1254	534	594	213	356	324	245	398	390	192	218	79	125	80	162	42	119	79	111	83
(11)	676	396	393	105	426	236	206	202	193	229	123	76	144	41	70	198	127	48	86	243
(12)	2166	3518	3478	2007	652	1357	1466	1869	1877	1289	1638	414	1181	780	714	1271	917	747	1082	323
(13)	474	8	22	58	213	75	12	5	1	6	12	24	60	35	2980	11	24	34	10	9
(14)	16675	2103	2526	3273	2079	1837	1660	1747	1382	1120	1337	945	1543	1322	680	984	1148	1215	783	839
(15)	948	351	325	399	308	186	214	329	237	224	333	114	138	612	98	23	201	187	284	52
(16)	2365	2864	3086	2208	1688	1330	1462	1991	1826	1256	1674	681	955	687	685	603	870	1046	1030	349
(17)	1272	1424	1476	1990	751	763	1170	889	851	897	787	160	411	1136	239	301	486	374	387	239
(18)	654	1211	1149	664	350	254	839	513	615	610	460	87	131	386	91	70	223	198	143	89
(19)	170	64	72	45	64	32	62	61	49	58	23	22	16	54	134	2	43	18	7	44
(20)	1083	233	313	161	184	268	216	94	105	180	83	30	85	26	78	104	41	11	55	537
(21)	444	352	242	92	69	124	144	120	119	160	55	26	49	28	78	64	39	13	38	189
(22)	528	81	46	78	29	32	57	76	34	74	19	8	33	27	37	8	22	13	21	130
(23)	2214	732	586	381	335	381	543	444	316	532	206	134	220	100	138	42	156	91	124	320
(24)	1305	864	645	194	298	324	452	311	269	420	134	66	187	41	116	146	131	34	104	293
(25)	580	484	431	142	389	408	322	250	208	285	95	48	170	70	87	276	110	23	140	80
(26)	1306	756	761	329	404	512	501	272	329	503	203	44	300	82	80	396	160	19	251	130
(27)	377	161	194	29	51	82	141	100	84	120	44	3	55	9	40	57	25	7	24	69
(28)	2158	967	852	423	324	630	550	565	462	531	251	137	281	197	251	133	213	75	211	718
(29)	1789	1405	1203	441	685	782	799	767	658	699	434	62	663	74	251	639	247	44	297	633
(30)	502	231	178	28	98	531	141	97	98	177	16	32	70	17	30	59	33	39	48	53
(31)	799	832	566	91	202	370	368	205	159	336	49	39	153	37	46	211	74	32	123	79
(32)	597	222	245	68	92	179	191	117	102	226	37	43	39	40	62	5	48	54	41	42
(33)	102	81	48	22	13	112	70	64	43	88	19	1	12	4	27	7	74	6	46	4
(34)	300	220	204	31	35	195	179	358	216	196	176	5	27	3	86	22	196	20	102	11
(35)	935	952	659	105	192	1011	459	341	232	416	69	54	166	46	159	122	123	76	135	91
(36)	2500	1731	1191	753	644	1275	744	872	716	807	365	201	451	183	221	485	281	311	436	145

In Figs. 5 and 6 we have shown pairs of countries mentioned together the most frequently for subsequent months for two selected sources of information: *Guardian* and *China Daily*. On the basis of Table 2 we can observe highest positions of countries according to scoring by selected complex network measures. For the comparison, we have chosen the following measures: *Degree*, *Weighted Degree*, *Closenes*, *Betweenness*, *Eigenvector*, *Clustering*, and *PageRank*. We can notice that USA has the best scores in all measures describing importance and influence in the network. The other countries showing high scores are: China, U.K., France and Russia.

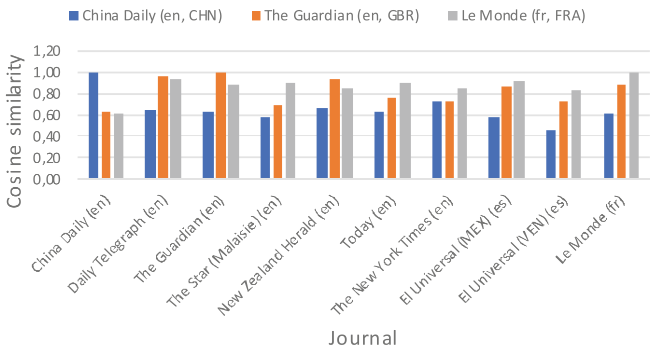


Fig. 3. Cosine similarity measure describing the countries mentioned in following time periods for selected journals

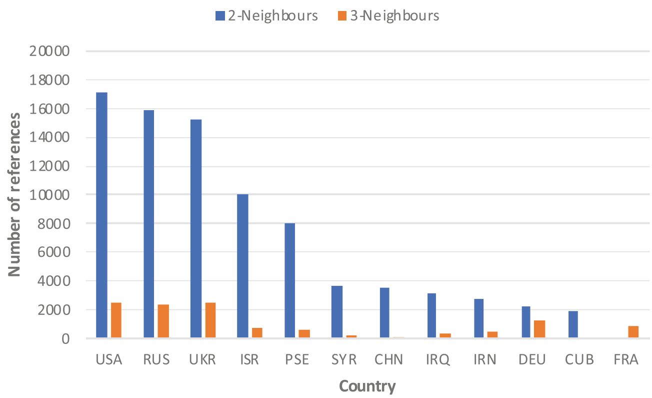


Fig. 4. Countries most frequently referenced together with other countries in groups consisting of 2 or 3 countries

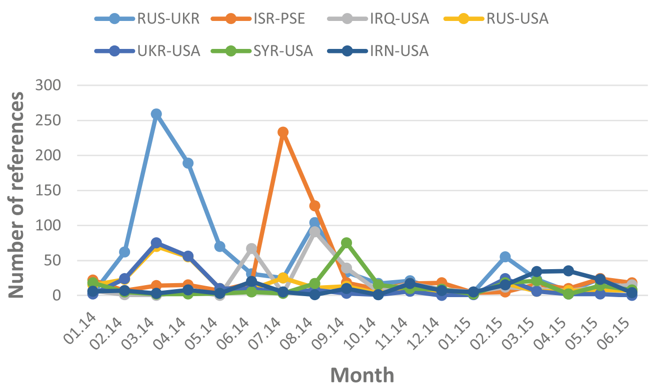


Fig. 5. Groups of countries mentioned together in the press notes of Guardian the most frequently, in following months

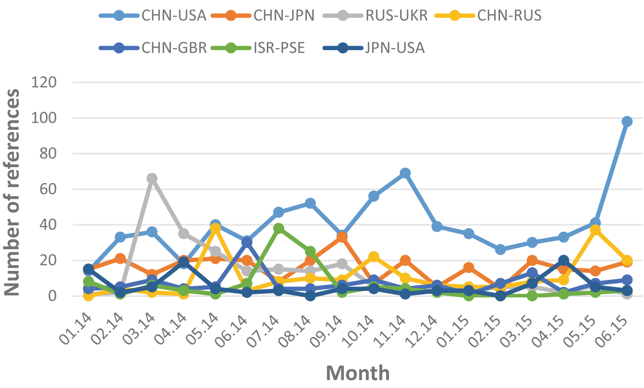


Fig. 6. Groups of countries mentioned together in the press notes of China Daily the most frequently, in following months

Table 2. Highest positions of countries in scoring: selected measures: (i) Degree, (ii) Weighted Degree, (iii) Closenes, (iv) Betweenness, (v) Eigenvector, (vi) Clustering, (vii) PageRank

Country	(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)
USA	164	2043	1.194	2966.755	1.000	0.207	0.026
CHN	153	1141	1.239	2462.059	0.959	0.222	0.024
GBR	137	1200	1.323	1382.345	0.941	0.269	0.021
FRA	131	1295	1.353	1352.106	0.913	0.280	0.020
ESP	107	615	1.483	781.629	0.814	0.343	0.016
AUS	91	581	1.552	780.509	0.726	0.386	0.015
RUS	114	1164	1.448	696.215	0.842	0.320	0.015
NGA	94	353	1.537	460.546	0.738	0.369	0.014
DEU	105	840	1.493	430.796	0.831	0.371	0.015
ITA	91	601	1.562	420.743	0.768	0.429	0.013

6 Conclusion

In the paper the model of agent–based system for filtering, integration and analysis of information representing selected notions and relations between them has been proposed. For this purpose we applied the multi-agent approach with techniques of data exploration and social network analysis. We focused on information about countries present in press notes published by popular international journals which were chosen as a test dataset.

In further works we plan to develop the model and the created system, to enrich mechanisms of information interpretation as well as the use of the other more universal sets of notions occurring in information sources.

References

1. Bellifemine, F.L., Caire, G., Greenwood, D.: Developing Multi-Agent Systems with JADE (Wiley Series in Agent Technology). Wiley, Hoboken (2007)
2. Demazeau, Y., Costa, A.R.: Populations and organizations in open multi-agent systems. In: 1st National Symposium on Parallel and Distributed AI (PDAI 1996) (1996)
3. Giraud, T., Grasland, C., Lamarche-Perrin, R., Demazeau, Y., Vincent, J.M., Thévenin, T.: Identification of international media events by spatial and temporal aggregation of RSS flows of newspapers. In: 18th European Colloquium in Theoretical and Quantitative Geography (ECTQG), pp. 112–114. Dourdan, France, September 2013
4. Gliwa, B., Koźlak, J., Zygmunt, A., Cetnarowicz, K.: Models of social groups in blogosphere based on information about comment addressees and sentiments. In: Aberer, K., Flache, A., Jager, W., Liu, L., Tang, J., Guéret, C. (eds.) SocInfo 2012. LNCS, vol. 7710, pp. 475–488. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35386-4_35
5. Gliwa, B., Koźlak, J., Zygmunt, A., Demazeau, Y.: Combining agent-based and social network analysis approaches to recognition of role influence in social media. In: Demazeau, Y., Ito, T., Bajo, J., Escalona, M.J. (eds.) PAAMS 2016. LNCS (LNAI), vol. 9662, pp. 109–120. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39324-7_10
6. Horling, B., Lesser, V.: A survey of multi-agent organizational paradigms. Knowl. Eng. Rev. **19**(4), 281–316 (2004)
7. Kozlak, J., Zygmunt, A.: Agent-based modelling of social organisations. In: Proceedings of the 2011 International Conference on Complex, Intelligent, and Software Intensive Systems, CISIS 2011, pp. 467–472. IEEE Computer Society, Washington, DC (2011). https://doi.org/10.1109/CISIS.2011.76
8. Lamarche-Perrin, R., Demazeau, Y., Vincent, J.M.: The best-partitions problem: how to build meaningful aggregations. In: 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), vol. 2, pp. 399–404, November 2013
9. 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 (LNAI), vol. 7879, pp. 157–169. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38073-0_14
10. Maes, S., Meganck, S., Manderick, B.: Inference in multi-agent causal models. Int. J. Approximate Reason. **46**(2), 274–299 (2007). Special Track on Uncertain Reasoning of the 18th International Florida Artificial Intelligence Research Symposium (FLAIRS 2005)
11. Newman, M.: Networks: An Introduction. Oxford University Press Inc., New York (2010)
12. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM **18**(11), 613–620 (1975)
13. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications, vol. 8. Cambridge University Press, Cambridge (1994)

Check for updates

Evaluation of Multi-agent Coordination on Embedded Systems

Marcelo S. Menegol(✉), Jomi F. Hübner, and Leandro B. Becker

Universidade Federal de Santa Catarina, Florianópolis, SC, Brazil
m.s.menegol@posgrad.ufsc.br, {jomi.hubner,leandro.becker}@ufsc.br

Abstract. Efforts have already been made to embed agents and agent-oriented technology into robots and other systems. For this purpose, many architectures and frameworks have been developed to allow software agents to interact properly with hardware components of theses systems. However, few proposals have evaluated a system where multiple agents are involved and require coordination. This paper compares the multi-agent systems approach, with a multi-agent-oriented programming language, against a more standard approach, with imperative programming, in the development of embedded systems that require coordination. To support the comparison, applications were implemented and evaluated with hardware-in-the-loop.

Keywords: BDI agents · Coordination · UAV · Embedded systems

1 Introduction

Embedded systems are a big part of people's daily lives. When people use an Internet modem, a printer or a phone, they are usually interacting with an embedded system. But besides those electronic products, the industry behind many consumer-goods also rely heavily on embedded systems, in the form of robots and other machinery. The adoption of robots in many fields have been on the rise for years now and, as it is to be expected for an expanding technology, an ever-growing range of technologies are being applied to the development of robotic systems. This includes Agent-oriented Programming (AOP) and Multi-Agent Systems (MAS).

Many embedded systems applications require (or benefit from) the use of many devices instead of just one to perform the same task. Consider, for example, the use of UAVs for search and rescue operations on disaster struck environments. To search an area and swiftly deliver supplies to the victims, a single UAV carrying both camera and supplies can be used. But the distance that an UAV can travel is inversely proportional to the weight it carries. So two UAVs, one with a camera and another just to deliver resources can provide a more efficient solution. A larger area can be covered if the scout UAV searches for victims and the courier UAV only flies directly towards the delivery locations.

The authors are grateful for the support given by CAPES (Pró-Alertas project).

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 212–223, 2018.
https://doi.org/10.1007/978-3-319-94580-4_17

Many of the applications that use multiple robots, as the one just presented, also require that the entities coordinate with one another. On the example provided, the scout must be able to communicate with the courier to warn where it should make its deliveries. The courier must not try to do a delivery without directions from the scout. To enforce these kind of restriction is to enforce coordination between the robots.

Diakopoulos and Cass [1] shows that the main languages used for embedded systems development are C and C++: essentially imperative languages. Books on embedded systems, such as [2], suggests that embedded systems are usually programmed with the imperative paradigm. When systems programmed on this paradigm need to coordinate themselves, message exchanging has to carefully be designed and added to the programs to guarantee the sequence of execution.

The multi-agent systems field has already explored many tools and techniques regarding coordination, as [3,4] points out. Some frameworks, such as JaCaMo [5], already provide tools to enforce coordination. To apply those to embedded systems, the devices need to be modeled and programmed as agents. This is not always a straight-forward task: agent frameworks are not always made to be embedded and to interact with the hardware required in these applications. It is also not clear yet how well these methods perform when compared to other approaches.

Thus, in this paper, an evaluation of multi-agent coordination applied to embedded systems is be presented. The hardware and software specification used in the experiments are described (Sect. 3), as well as the application that inspired the research. Challenges pertaining to the development of the agent-oriented approach are presented (Sect. 4), as well as two experiments involving UAVs (Sect. 5). Then, an agent-oriented solution to the problem is compared to a solution deemed to be more traditional, with imperative programming (Sect. 6). The advantages and drawbacks of using each approach for embedded systems are discussed (Sect. 7).

2 Related Works

In the MAS field, the adoption of agent-oriented programming on autonomous robots is not a novel idea. The paradigm has been applied in many different ways. Some have modeled single robots as multi-agent systems (MAS). Others have proposed frameworks and tools for programming robotics systems. Most of the works were tested on simulation only, while some others have been validated with a hardware-in-the-loop approach.

There are many tools and frameworks that enable a computer agent to control or interact with robotics systems. Wei and Hindriks [6] have developed an architecture that allows GOAL [7] agents to interact with a robotic hardware. This research was focused on mobile robots, and thus had features specific to mobile vehicles applications, such as navigation-related modules. This architecture mainly consisted of a middleware that connects the high-level layer (the agent) with a low-level (hardware control). This modularity is a design pattern that has repeated itself in other frameworks. The architecture was then tested on the humanoid robot Nao [8] in an indoor environment.

For the AgentSpeak interpreter Jason [9], there is an architecture called ARGO [10]. It consists of a middleware that enables Jason's agents to communicate with low-level hardware (micro-controllers) by using a library called Javino [11]. In this context, the agent is assumed to be executing on a machine with considerable performance, such as a personal computer, and performing actions or receiving percepts through another lower-level hardware, such as an Arduino, which is actually in charge of the actuator or sensor.

While ARGO assumes that the low-level control happens in a separate, lower-level hardware, JaCaROS [12] is a tool that connects the cognitive agents of Jason to the Robot Operating System. ROS is a very versatile tool for robot control and simulation. JaCaROS then provides a framework that connects the agent to either a real environment (through hardware control) or a simulated one: as long as the machine is capable of running ROS.

Fichera and colleagues described PROFETA (Python RObotic Framework for dEsigning sTrAtegies) [13]. The framework uses the cognitive agent model and implements AgentSpeak(L) in Python. This framework, being developed in a monolithic platform, does not have the same layered architecture as others have. The author argues that there is one big advantage in using an integrated framework: the developer does not have to deal with different execution environments, one for low and other for high-level logic, making the system easier to debug and maintain.

SOIFRA [14] is a framework for mobile robot autonomy. It allows for JADE [15] agents to control the robot on a behavioral level. The framework already has object detection and collision avoidance built into it and accepts other independent algorithms. The software provides modularity through a service-oriented architecture.

Santos et al. [16] has shown that it is possible to use Jason on an embedded system (BeagleBone Black board) in a UAV context. The result of this work has shown that there are advantages of using AOP to program UAV applications when compared to, for example, the imperative paradigm (using C), a more traditional approach. Namely, the main advantages were on the size of the program (size in Kbytes of the source files) and on how easy it is to imbue pro-activity into the program and to modify its behavior later on. Among the disadvantages, there is an increase in CPU usage for the agent-oriented approach, when compared to a traditional one.

The agent's framework publications presented so far are not a classified here according to four criteria, identified by the following questions:

– Are they tested on embedded systems?
– Do they use an agent oriented programming language?
– Do they cope with more than one agent?
– Do they use a coordination tool?

Table 1 presents the classification results. As the table shows, none of the aforementioned publications address the issue of coordination between multiple

Table 1. Framework comparison table.

Framework	Embedded?	AOP language?	Multi-agent?	Coordination?
GOAL	Yes	Yes	No	No
ARGO	Yes	Yes	No	No
JaCaROS	No	Yes	Yes	No
PROFETA	No	No	No	No
SOIFRA	No	No	Yes	No
Jason, by Santos	Yes	Yes	No	No

agents by default. There are no ready-made tools for coordination. This aspect, if implemented, would rely on an ad-hoc solution. In tour proposal, all of the criteria presented are met. This project was developed and tested with hardware-in-the-loop, uses an agent oriented programming language, and addresses multi-agent coordination.

It is worth noting that there is a lack of publications that compares the AOP approach to the traditional approach for embedded systems. One of the reasons is that any work that proposes to make such evaluation needs to implement a solution in both paradigms. Implementing a solution in two different paradigms might be too cumbersome.

3 Embedded Architecture

The applications developed and evaluated in this paper intend to be compatible with real UAVs. To embedded an agent in an aircraft, an embedded computer (EC) has to be used along with the flight controller (FC). The chosen hardware for that purpose is the BeagleBone Black [18] single board computer, with 512MB of RAM, running Debian operating system. For the agent to command the vehicle, packets are sent to the FC using the MAVLINK [19] protocol. EC and FC are connected via serial interface. The DRONEKIT package [20] is used to handle the MAVLINK messages.

This architecture was tested and validated on a single real UAV: a SK450 assembly with a Pixhawk as FC. For the purposes of this paper though, which requires many UAVs flying together, the aircrafts are simulated in Ardupilot's software-in-the-loop simulator [17]. It simulates the behavior and interface of flight control boards available on the market.

For simulation purposes, a Python program simulates a wireless device, routing messages to and from the agent to another Python program that properly dispatches massages to other agents, simulating a mesh network in a wireless environment. Both programs were developed for this work specifically.

4 Agent-Hardware Integration

The agent-hardware integration used in this project is a middleware that allows Jason [9] agents to run embedded applications. The main purpose of this middleware is to give the agents access to the hardware's interfaces to peripherals, which Jason (or JaCaMo [5], for that matter) does not provide natively. Instead of re-implementing that, the focus was on developing an architecture that could communicate with other programs, written in other languages, like C or Python, that could then access the hardware when needed.

As already proposed by others (Sect. 2), this middleware is divided in two main pieces: one for handling the high-level logic, and another for the low-level. For ease of understanding, those will be called *head* and *body* from now on. The *head* itself is divided in two pieces: the *cortex*, a Java class that extends and overwrites some Jason agent's methods, and the *bulb*, a Java class that handles the interface with the *body*. The *body* is loosely defined as a software that performs the actions that the agent requires and sends information from sensors as percepts to the agent. For the purpose of the UAV application here presented, it was written in Python.

In this architecture, the agent launches a thread during initialization that connects itself, via socket, to the *body*. The *body*, which is run beforehand, functions as a server, waiting to accept this connection. The information flow is depicted in Fig. 1: information comes as percepts from the environment or as a message from another agent, goes through the *body* to the *cortex* via the *bulb*; Then, after the agent's reasoning cycle, it flows back to the *body* as a message to be sent to another agent or as an action to be performed. The *body* then executes the will of the agent, either by sending the message or executing the action.

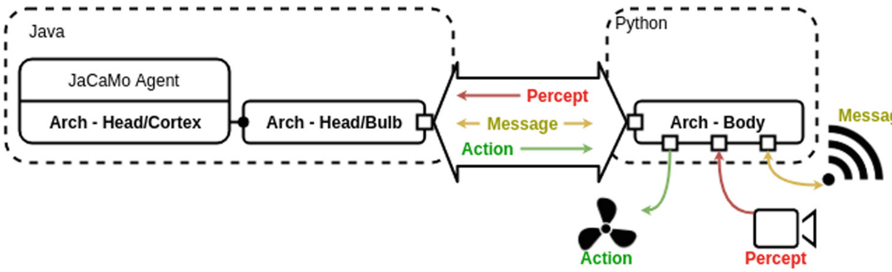


Fig. 1. Information flow in the agent-hardware integration architecture.

4.1 Coordination Platform

One of the great strengths of using agent technology is on how it handles coordination. The JaCaMo framework allows the development of applications where coordination is paramount by using the Moise [21] organizational model. On this framework, each agent takes responsibility of a set of tasks and the platform makes sure that the tasks are executed in the correct order.

Moise is a middleware that comes integrated with JaCaMo. It "is the interface between the agents and the overall system, providing access to the communication layer [...], information about the current state of the organization (created groups, schemes, roles assignments, etc.), and allowing the agents to change the organization entity and specification" [21].

To make use of Moise in JaCaMo, a few things have to be provided. First, there is the organizational specification: a file that describes the intricacies of the organization that the agents will try to conform to. It is defined by three sub-specifications: structural, functional and normative. Each of them serve an specific purpose to coordinate groups of agents.

Since this work focuses on the execution of tasks, only the functional specification, that is focused directly on those tasks, is relevant. The functional specification defines plans and goals that the agent should try to achieve, as well as their characteristics: how many agents each goal takes, in what order they need to be performed, if the goals can be divided into missions etc.

According to [21], the functional specification is composed of a collective *global goal* and "how these goals are decomposed (by plans) and distributed to the agents (by *missions*)". In this context, *plans* are descriptors of relationships between sub-goals. If certain goals need to be accomplished in a sequence, for example, this is described by a plan. *Missions* are sub-sets of goals. The agent that commits to a given mission should pursue all the goals in that mission.

Whenever an application is launched in JaCaMo, any agent can create an organization by providing the specification as argument. Then, they can create an artifact called *scheme board*, which can be used by the agents to monitor the organization and the progress of the missions.

During execution, agents can try to commit to the missions described in the organization. The scheme board checks if the mission is compatible with the request (e.g., if there is still room for more agents), and then registers the commitment or fails the action. The scheme board then will track the progress of the missions and warn agents when they should start pursuing each goal, ensuing the correct execution of the plans. In short, this is how agents interact with Moise to perform missions in a coordinated fashion.

Usually, this platform works in a centralizing manner: there is one shared instance of the scheme board (for each common scheme) that must oversee all committed agents, and that those agents must report to. In this case, all the agents are executing in the same virtual environment. They all communicate with a single application executing on a single machine, where the scheme board was created.

To use the coordinator software on distributed mobile robot's applications, where communication is not always too reliable, distributed schemes are used. Instead of all robots reporting to the same scheme board, each robot has its own. By incorporating a set of behavior to the agents, they become able to share schemes and merge other schemes with their own. This allows for the scheme board to update the state of its own scheme by using other schemes. Figure 2 illustrates a situation in which two agents only have information about their own commitments to missions and what happens after one agent receives and merges the scheme of the other.

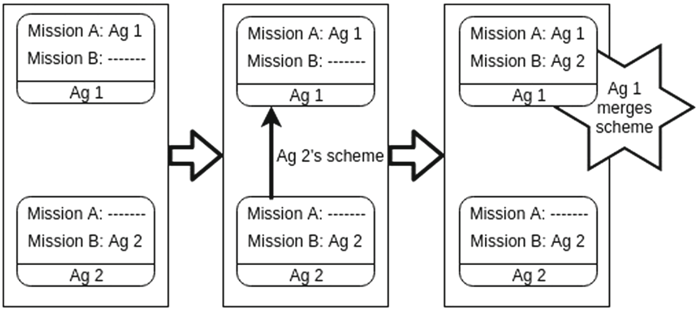


Fig. 2. Scheme sharing and merging in action.

In this approach, each agent shares the same initial scheme, by having the same organizational specification file at initialization. They periodically (the period depends on how dynamic the application is) share their whole schemes and merge. This way, although separated, they manage to keep the same state of the scheme. The vehicles do not have to maintain a stable connection at all times among themselves. There is no central entity that they must report to and no single point of failure.

With this approach, the JaCaMo platform maintains the distributed state of the scheme available to the agents and inform them whenever their goals can be pursued. The programmer of the agents just need to provide plans to achieve the goals.

5 Application Development

To verify the effectiveness of the framework and coordination strategy, and compare it with the imperative approach, two tests were performed. The first is a simple coordination example, made to assert if the whole system has its basic functionality working correctly, with two simulated UAVs. The second is a more elaborate test to verify if the strategy is capable of handling more complex scenarios, also with two UAVs.

Besides the UAV agents, another agent was developed: the *manager* agent. The purpose of this agent is to give some control and monitoring capabilities over the experiment to the human staff. It is responsible for initially messaging the UAVs so they can start execution, and initially distributing missions. The progress of the missions can be monitored with this agent, since the agents will occasionally share their progress with it.

To evaluate the adopted strategy, the same applications were also reproduced with imperative programming paradigm, in C. Each UAV had its mission hard-coded in their program, with cascading if-then-else control structures. To ensure coordination, simple message exchanging was performed periodically between the UAVs.

Due to space constraints, we do not present here implementations' details. The source code used though can be downloaded from:

https://github.com/msmenegol/UAVExperiments.

5.1 Simple Coordination Experiment

Two UAV agents were used for this experiment, with one of them running on the EC. They both have the same software configuration, with the exception of their Jason agent names, which were *alice* (A) and *bob* (B).

The application consists of two UAVs that fly to a common geographical spot P in a sequence determined by their organizational model. There are two missions, *Ma* and *Mb*, each requires one and only one agent committed to it. *Ma* consists of two goals: take off and fly to point P (*G1*), go back to the take off site and land (*G3*), both at a 10 m height. *Mb*'s goals (*G2* and *G4*) are much like *Ma*'s, but the height is set for 20 m. Those goals are structured in a sequence, from G1 to G4, so that one cannot begin until the previous one was accomplished. A graphical representation of the scheme can be seen on Fig. 3 The difference in height was defined to make sure the UAVs would not intercept one another mid-air on a real scenario.

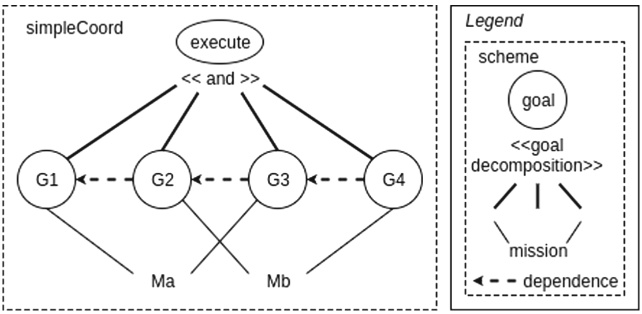


Fig. 3. Representation of the simple coordination scheme.

Execution. As soon as the manager agent commands the UAVs to commit to their respective missions, they each commit and share their scheme with one another. Then, agent A (with mission Ma) flies to P at a 10 meters height. Meanwhile, agent B does nothing, since its goals (G2 and G4) cannot be pursued yet. When A reaches P, goal G1 is achieved and A sends its scheme to agent B and stays put. The goal G2 is now enabled (since its dependence to G1 has been fulfilled) and B starts to pursue it. Thus, agent B flies to P, at a higher height. When it reaches P, goal G2 is achieved and B sends its scheme to A and stays hovering over P. The goal G3 becomes enabled and A starts to pursue G3.

Agent A goes back to where it took off and sends its scheme to B. Similarly, B goes back to where it took off and warns A. With both missions terminated and with the global goal achieved, they stop the motors.

5.2 Mock Lifebuoy Delivery Experiment

This experiment is a mock of a search and rescue operation described in Sect. 3. No real cameras or lifebuoys were used for this experiment. Instead, the scout (S) had three way-points already registered as initial beliefs. Instead of checking for real images, it only compares if it get close to any of the mock rescue points. When it does, it sends a message to the courier UAV (C). C, instead of making a delivery, only moves to the location it received from S and then goes back to base. The way-points and beliefs signaling the equipment that each drone had (imaginary camera or robotic gripper) were provided by the organizational specification. For this experiment, S ran on the BeagleBone board. The other agent ran on a common desktop computer.

Execution. The manager starts the experiment by commanding both UAVs to commit to their respective missions. S then takes off and flies by pre-determined way-points. When it gets within a 3 m radius to the first rescue point, it sends a message to C, without stopping. C then takes off, flies to the point and, when it gets within a 3 m radius to it, makes a delivery, setting its amount of buoys carried to zero.

After C made its first delivery S reaches the second rescue point. A message is sent again to C, but it cannot make another delivery without going back to its landing spot. C returns, lands (to symbolize a buoy reload), takes off again and goes to the second spot to make its delivery.

When S reaches the third point, C has not made its second delivery yet. After receiving the third message, it registers the third point but keeps on with the second one. Right after the second delivery is made, it immediately returns to the landing spot to recharge and make its third one. S goes back to its landing spot, concluding its route, which means that there should be no more delivery points to be found. After C has concluded its third delivery, it goes back as well. The global goal is accomplished, since S has finished its route and C has no more deliveries to make.

6 Evaluation

Both experiments were perform with both the agent-based and the imperative approaches three times. Memory usage was measured system-wide. CPU usage was measured only on the user processes. Both were measured every 5 s. The experiments vary in duration, but none went over 7 min. The performance was measured on the UAV executing on the BeagleBone on all cases.

To measure the size of each approach, the source files developed specifically for each application were compressed using the *gzip* compression utility. This diminishes the influence of blank spaces and line breaks that are usually used in software programs to improve readability.

6.1 Simple Coordination Experiment

Both the agent-based and the imperative approaches succeed in performing the simple tasks in the right order. Table 2 shows the results. The memory usage is higher with JaCaMo (119% of the imperative approach), but CPU usage is lower (65%), on average. Program size is also smaller (79%).

Table 2. Simple coordination experiment - results

Applications	CPU usage (% time)				Memory usage (% allocated)				Size (KB)
	Avg.	Max.	Min.	Std.	Avg.	Max.	Min.	Std.	
Agent-based	21.13	29.81	6.17	6.24	81.99	84.48	56.05	4.55	5.894
Imperative	32.69	37.16	26.92	2.91	68.66	68.75	66.33	0.32	7.414

6.2 Mock Lifebuoy Delivery Experiment

On this experiment, the BeagleBone was executing the scout agent. The results can be seen in Table 3. CPU and memory usage is 4% and 21% higher with the agent approach. Program size is 81% of the imperative approach. Both implementations executed as expected.

Table 3. Mock lifebuoy delivery experiment - results

Applications	CPU usage (% time)				Memory usage (% allocated)				Size (KB)
	Avg.	Max.	Min.	Std.	Avg.	Max.	Min.	Std.	
Agent-based	37.47	40.09	34.93	1.32	83.24	84.48	66.33	2.98	6.626
Imperative	35.71	38.33	31.10	2.83	68.70	68.75	66.33	0.27	8.191

A second (more subjective) analysis can be made regarding the programming effort needed to transition between the first and the second experiment. On AgentSpeak, the addition of new logic consists mostly of adding new behaviors or plans to the agent's code. New behaviors seldom interact with previous ones, so very few changes on the existing code has to be made to accommodate them. Due to the structured nature of C, adding new logic more than often means that the existing logic has to change. Moreover, a finite state machine had to be programmed in C to keep track of the state of the application. As more entities participate in the application (in this case, UAVs), the state machine would have to grow to keep track of more possible states. This might undermine the scalability of the application.

With AgentSpeak on JaCaMo, the same plans regarding moving toward a way-point were mostly preserved. The actions in the underlying integration software remained exactly the same. More behaviors and beliefs had only to be added. Almost no modifications had to be done on the already existing agent's behaviors. Most of the changes were on the organizational model, which had to be re-written to describe the new application.

On the other hand, the C code had to be re-written almost entirely. A different order in the execution of actions translates almost directly to a different order in the program structure. This might make changing from one application or mission to the next much harder.

7 Conclusion and Future Works

The first conclusion that can be drawn from these experiments is that it is possible to embed JaCaMo agents to command robots. More importantly, JaCaMo's tools for coordination can also be used in that setting. This opens up the scope of possible applications. With JaCaMo embedded, robots can not only perform complex behavior and reasoning, but also coordinate with other robots seamlessly.

Although the imperative approach might consume less computational resources than the agent-based one, both run on similar levels. Memory and CPU usage does not go over 85% and 45% respectively with the agent-based approach. These results reinforce the notion that, although agent-based solutions for embedded systems might require more computational power to execute, they are not prohibitive, and are a small price to pay for the gain in ease of programming.

The presented experiments relied on a manager agent to distribute missions. In the future, better methods for properly distributing missions among agents should be developed. It would also be interesting to study how to properly coordinate planning agents, that might re-arrange their actions based on the conditions of the environment.

We also plan to improve the comparison against other approaches measuring the effort a set of programmers require to implement applications like those discussed here. As many practical issues might appear when moving from simulation to real UAVs, such effort is also interesting to be addressed as future work.

References

1. Diakopoulos, N., Cass, S.: Interactive: the top programming languages 2016. In: IEEE Spectrum: Technology, Engineering and Science News, July 2018. https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2016
2. Lee, E.A., Seshia, S.A.: Introduction to Embedded Systems: A Cyber-Physical Systems Approach, 2nd edn. MIT Press, Cambridge (2017)
3. Weiss, G.: Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press, Cambridge (1999)

4. O'Hare, G.M.P., Jennings, N.R.: Foundations of Distributed Artificial Intelligence. Wiley, Hoboken (1996)
5. Boissier, O., Bordini, R.H., Hübner, J.F., Ricci, A., Santi, A.: Multi-agent oriented programming with JaCaMo. Sci. Comput. Program. **78**(6), 747–761 (2013)
6. Wei, C., Hindriks, K.V.: An agent-based cognitive robot architecture. In: Dastani, M., Hübner, J.F., Logan, B. (eds.) ProMAS 2012. LNCS (LNAI), vol. 7837, pp. 54–71. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38700-5_4
7. Hindriks, K.V.: GOAL Programming Guide. Delft University of Technology (2011)
8. Gouaillier, D., Hugel, V., Blazevic, P., et al.: Mechatronic design of NAO humanoid. In: Proceedings of ICRA 2009, pp. 769–774 (2009)
9. Bordini, R.H., Hübner, J.F., Wooldridge, M.: Programming Multi-Agent Systems in AgentSpeak Using Jason. Wiley Series in Agent Technology. Wiley, Hoboken (2007)
10. Pantoja, C.E., Stabile, M.F., Lazarin, N.M., Sichman, J.S.: ARGO: an extended jason architecture that facilitates embedded robotic agents programming. In: Baldoni, M., Müller, J.P., Nunes, I., Zalila-Wenkstern, R. (eds.) EMAS 2016. LNCS (LNAI), vol. 10093, pp. 136–155. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50983-9_8
11. Lazarin, N.M., Pantoja, C.E.: A robotic-agent platform for embedding software agents using raspberry pi and arduino boards. In: 9th Software Agents, Environments and Applications School (2015)
12. Wesz, R.B., Meneguzzi, F.R.: Integrating robot control into the Agentspeak(L) programming language. In: Pontifícia Universidade Católica do Rio Grande do Sul (2015)
13. Fichera, L., Messina, F., Pappalardo, G., Santoro, C.: A python framework for programming autonomous robots using a declarative approach. Sci. Comput. Program. **139**, 36–55 (2017). ISSN 0167–6423
14. Arokiasami, W.A., Vadakkepat, P., Tan, K.C., et al.: Interoperable multi-agent framework for unmanned aerial/ground vehicles: towards robot autonomy. Complex Intell. Syst. **2**, 45 (2016)
15. Bellifemine, F., Poggi, A., Rimassa, G.: JADE: a FIPA2000 compliant agent development environment. In: Proceedings of AGENTS 2001, pp. 216–217. ACM (2001)
16. Santos, F.R., Hübner, J.F., Becker, L.B.: Concepção e análise de um modelo de agente BDI voltado para o planejamento de rota em um VANT [Conception and analysis of a BDI agent model for UAV's path planning]. In: Proceedings of WEESAC 2015. UFF (2015)
17. ArduPilot Development Team: SITL Simulator, January 2018. http://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html
18. Coley, G., Kridner, J.: BeagleBone Black System Reference Manual, January 2018. https://github.com/beagleboard/beaglebone-black/wiki/System-Reference-Manual
19. Dronecode Project: MAVLink Micro Air Vehicle Communication Protocol, January 2018. http://qgroundcontrol.org/mavlink/
20. 3D Robotics: DroneKit-Python's documentation, January 2018. http://python.dronekit.io/
21. Hübner, J.F., Sichman, J.S., Boissier, O.: $\mathcal{S}-\mathcal{M}oise^{+}$: a middleware for developing organised multi-agent systems. In: Boissier, O., Padget, J., Dignum, V., Lindemann, G., Matson, E., Ossowski, S., Sichman, J.S., Vázquez-Salceda, J. (eds.) AAMAS 2005. LNCS (LNAI), vol. 3913, pp. 64–77. Springer, Heidelberg (2006). https://doi.org/10.1007/11775331_5

Check for updates

AgentUDE17: A Genetic Algorithm to Optimize the Parameters of an Electricity Tariff in a Smart Grid Environment

Serkan Özdemir[1(✉)] and Rainer Unland[1,2]

[1] DAWIS, University of Duisburg-Essen, Schützenbahn 70, 45127 Essen, Germany
{serkan.oezdemir, rainer.unland}@icb.uni-due.de
[2] Department of Information Systems, Poznan University of Economics, Poznan, Poland

Abstract. Electricity retailers are the most vulnerable actors in the electricity grid since they are responsible for many financial challenges. As a business entity, electricity retailers aim to maximize their sales volume and minimize the procurement cost in a highly competitive environment. On the retail market side, they publish rich tariffs specifications, which resolves the needs of customers in time, energy, and financial domains. In the paper, we present an online genetic algorithm that optimizes the parameters of an electricity consumption tariff, such as unit retail price, periodic, sign-up, and early withdrawal penalty payments on the fly. Additionally, we use time-of-use (TOU) price scheme to reduce peak-demand charges. The algorithm was first deployed and tested in our winning broker agent (AgentUDE17), which competed in the Power Trading Agent Competition (Power TAC) 2017 Final games. In the paper, we first present the theoretical background and detail the concepts of the algorithm. Secondly, we comparatively analyze the tournament data. Post tournament analysis shows that AgentUDE17 successfully optimized its tariff parameters on the fly and significantly increased its utility. Additionally, it managed to reduce its peak-demand charges thanks to its adaptive TOU price schemes.

Keywords: Autonomous agent · Retail market · Optimization Genetic algorithms

1 Introduction

Energy transition policies have been challenging the intermediary power actors (hereinafter broker and retailer are used interchangeably) since they take the burden of increasing costs and penalties on the changing energy landscape [7]. The smart grid can help both customers and brokers for an effective management of resources in real-time [3]. In that sense, autonomous smart algorithms can be employed by these entities to make full use of available data. Before using such smart algorithms in the real-world, competitive agent-based simulations are needed to identify the destructive and unanticipated behaviors of the grid participants. We use Power Trading Agent Competition

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 224–236, 2018.
https://doi.org/10.1007/978-3-319-94580-4_18

(Power TAC) to simulate an autonomous power trading agent (AgentUDE17). In the paper, we only focus on the retail trading problem of the broker.

Brokers must publish rich, flexible tariff specifications to meet the needs of most customers, such as electric vehicles, local producers, households, and industrial consumers. A wider set of tariff parameters (i.e. unit retail price, periodic, sign-up, early withdrawal penalty payments, and contract length) makes the tariff specification more flexible and attractive. However, in terms of finding the optimal parameter set, many machine learning algorithms fail to explore the full search space, and, in most cases, they find a local optimum instead of a global optimum. In the paper, we present a genetic algorithm to find a parameter set (hereinafter chromosome or individual), which maximizes the net profit over a period. Results show that AgentUDE17 managed to gain significant profit, despite a tough competition. Moreover, peak-demand charges are significantly reduced thanks its TOU price scheme. Note that the peak-demand charge refers to the amortized cost of infrastructure, which billed weekly to the brokers in the extent of energy usage in peak hours.

The structure of the paper is organized as follows. Related work is explained in Sect. 2. Game description and definitions are given in Sect. 3. Our genetic algorithm is detailed in Sect. 4. Afterwards, Sect. 5 describes the tournament data and the analysis process as well as a section, dedicated to the results. Finally, the paper is concluded in Sect. 6 with an outlook to our future work.

2 Related Work

Many agent-based simulations exist to address the challenges in the smart grid applications (e.g. microgrids, demand response). In the field of competitive markets, Power TAC is one of the leading frameworks, which enables the competitive benchmarking in the smart grid [5].

Within the Power TAC framework, autonomous power trading has been studied by many research teams. Chowdhury et al. [2] use a Q-learning mechanism in the retail market, which rewards the gain or penalizes the loss in the profit. Given a state, their broker acts by incrementing, decrementing or keeping the tariff price. A winner broker Maxon16 [10] uses a TOU price scheme and determines the tariff rates using a Hill Climbing Algorithm. Another winner broker TacTex13 [11] uses a lookahead policy, which estimates the long-term utilities of candidate actions, and, in turn, reduces the complexity of searching in a high dimensional state-action space. Hoogland and Poutré [4] use a heuristic algorithm to determine the tariff price based on competitors' tariff prices. Wang et al. [12] use three independent SARSA processes to determine the tariff prices for 3 customer groups. Like in [10], the algorithm finds the optimal price by means changing the tariff price. Rúbio et al. [9] use a fuzzy logic algorithm based on their customer portfolio. They claim that the algorithm allows adaptive configurations, given a complex parameter set in the environment.

All approaches above fail to explore the full search space, which means that their algorithms focus on a small portion of the retail trading problem. In the best case, these algorithms end up with a local optimum in the search space. We aim to find a point near global optimum, taking all important tariff parameters into account. For sure, too many parameters result in a huge search space. Therefore, we use a genetic algorithm to overcome this problem. Genetic algorithms (GA) are used in many optimization problems where the problem cannot be solved easily due to the number of parameters and constraints added to the objective function [6]. Autonomous power trading is one of those applications that GA can conveniently be applied.

3 Game Description and Definitions

Power TAC [5] is an open-source simulation framework, which simulates competitive electricity markets for a robust policy guidance on the operation and structure of markets. A Power TAC broker is challenged to increase its profit by competing against other brokers and trading in the following energy markets:

- *Wholesale market* is a multi-channel spot market where buyers and sellers place their buy and sell orders for the next 24 delivery hours to match the customer demand. These auctions independently run in parallel and form a periodic double auction.
- *Balancing market* is operated by a distribution utility (DU) which ensures that the system is balanced and financially settled in real-time. DU may reward or penalize the brokers in the extent of their contribution or harm to the balancing process.
- *Retail Market* is the marketplace where brokers publish tariff specifications to attract customers. Customers are small and medium-sized consumers, producers and prosumers, and they may pick any tariff based on their economic preferences.

In Power TAC, time proceeds as a discrete time block, called time slot. A Power TAC game takes up to a random time slot count, starting from one, where the first 360 time slots are dedicated for a bootstrap period and the proceeding time slots for the competition. A tournament consists of multiple games, which are categorized in different game sizes, e.g. 3-player, 5-player, and 8-player games. Theoretically, each game is independently played. However, brokers can pass data from one game to another.

In a game, AgentUDE17 has only one active consumption tariff. Whenever a modification is needed during the optimization process, it supersedes the tariff with a new specification. In the paper, we will denote a tariff specification as $\tau = \langle p_{avg}, \grave{p}, bp, ewp, cl \rangle$. The parameters are now defined in detail:

- Tariff rates $\{p_1, p_2, \ldots, p_R\}$ (€/kWh): Unit cost of customer consumption. We use a single average rate p_{avg} to denote all rates (see Formula 2). Details to create a TOU scheme is detailed at the end of the next section.
- Periodic payment $\grave{p}$ (€): Daily payment (mandatory) from customers.
- Sign-up bonus bp (€): It is paid unconditionally by broker to subscribed customer.
- Early withdrawal penalty ewp (€): It is charged to customers, if they leave the tariff earlier than specified in the contract length.
- Contract length cl (in time slot, $cl \in N$): Minimum period in which the customer should stay in the tariff. If a customer subscribes to a tariff at time slot t and later withdraws it before time slot $cl + t$, then the customer pays a penalty (ewp).

Brokers may publish, revoke or modify tariffs anytime during the competition. In contrast to the new tariff publication, any modification on the tariff voids the early withdrawal penalty and contract length for existing customers. At any time slot, the tariff is published in the retail market, in the form of $\langle\{p_1, p_2, \ldots, p_R\}, \grave{p}, bp, ewp, cl\rangle$ and become available to subscriptions. Hereinafter, we call this tuple as a chromosome and individual interchangeably, which is denoted by C.

Definition 1 (Fitness Function). *Let C be a chromosome of an active tariff, between time slots t and $t+\delta$. Then the **fitness function** $f(C)$ returns a fitness value f_C, which is the total profit of the chromosome over a period. The function is defined as $\sum_t^{t+\delta} R_t - M_t$, where R_t is the net revenue (including $\grave{p}, bp, ewp$ and peak-demand charges) from the customers and M_t is the total of wholesale market commitments.*

In the fitness function, we excluded all other costs, such as metering fee, imbalance cost, tariff publication and revocation fees, since these variables may slightly change during the game and challenge the algorithm.

Definition 2 (Assessment Period). *Let τ_{ude} be an active tariff, published at time slot t. Then, assessment period δ is the time slot gap between time slots t and $t+\delta$. At time slot $t+\delta$, the active tariff is replaced by another tariff, $\tau_{ude} \leftarrow \tau_{ude}^{t+\delta}$. The length of the period is proportional to the tariff publication fee and may take values as $\delta \in \{42, 84, 168\}$.*

Definition 3 (Population). *Population is a set of N individuals $\{C_t^1, C_t^2, \ldots, C_t^N\}$ at time slot t, where the chromosomes (i.e. individuals) are ordered by their fitness values (i.e. $f_{C_t^1} \geq f_{C_t^2} \geq \ldots \geq f_{C_t^N}$). The population represents the collection of published and assessed tariffs in the past. The first chromosome represents the healthiest individual, which literally means that it has the highest fitness value. If a new chromosome satisfies the condition, $f_{C_t^N} \leq f_{C_t^{new}}$, then it is replaced with the least successful individual C_t^N.*

Definition 4 (Bootstrap Data). *Bootstrap data (BD) is an initialization database, which contains the initial population and tariffs of competitors in the form of $\{C_{BD}^1, C_{BD}^2, \ldots, C_{BD}^N, \tau_{B_1}, \tau_{B_2}, \ldots, \tau_{B_i}\}$ where τ_{B_i} is the opening tariff specification of i-th broker in the game. This tariff specification reflects the most recent data.*

We use a 3-level decision tree to store multiple bootstrap data. Top level BD stores the data from all games. Second level BD's (i.e. children of the top-level BD) are grouped by the game sizes, e.g. 3-, 5- and 8-players. Level 3 consists of unique broker sets, given the number of competing brokers and game size. For example, 5 brokers yield 10 unique combinations of possible games in 3-player mode. At the beginning of a tournament, AgentUDE17 has only one top-level database (BD). However, AgentUDE17 always requests a *BD* on the third level. If it does not exist on the third level, it firstly checks whether a *BD* exists on the second-level. If it is found, it is copied to the third level. Otherwise, the top-level *BD* is copied to both second level and third levels. After each assessment period, *BD* is updated on the fly, as the broker explores new individuals between two consecutive assessment periods. This can be formalized as $BD \leftarrow \{C_t^1, C_t^2, \ldots, C_t^N\}$. However, its parent and top-level databases are also affected from the update, so that their chromosome values are exponentially smoothed with empirically found weights 0.2 and 0.1, respectively. Likewise, competitors' opening tariffs are carried to the upper levels. The second and the top-level databases contain the most recent opening tariffs of all brokers in their sub-trees. This way, AgentUDE17 always gets the initial data about its competitors.

Definition 5 (Baseline Price). *Let τ_i be the i-th active tariff in the retail market at time slot t. Then the baseline price is $p_{base} = \min_{\tau_i \epsilon \tau} p(\tau_i)$, which literally means the cheapest active tariff in the market.*

However, the baseline price does not denote the lowest tariff price only, instead, $\grave{p}, bp, ewp$ and cl values of the respected tariff are also taken into consideration. We formalize the process using the formula (1):

$$p(\tau_i) = p_{avg}^{\tau_i} + \begin{cases} (\grave{p}^{\tau_i} - bp^{\tau_i})/\max\left(\delta_{day}, cl^{\tau_i}\right), & p_{avg}^{\tau_i} \leq \min\limits_{\tau_k \epsilon \tau, k \neq i} p(\tau_k) \\ (\grave{p}^{\tau_i} - bp^{\tau_i} + ewp^{\tau_i})/\max\left(\delta_{day}, cl^{\tau_i}\right), & else \end{cases} \tag{1}$$

Where δ_{day} refers to the hours in a day (=24) and $p_{avg}^{\tau_i}$ is the weighted mean of all rates in the tariff.

$$p_{avg}^{\tau_i} = \sum\nolimits_{r=1}^{R} p_r^{\tau_i} \omega_h \tag{2}$$

Where $\sum \omega_h = 1$ and h corresponds to the time interval in which the rate $p_r^{\tau_i}$ is active. The weight ω_h refers to the share of total distribution (grid wise) in the regarded interval. The impact of the bonus payment and early withdrawal penalty are projected to a single time slot and added to the average rate. If the average rate is the cheapest among all other tariffs, then we omit the impact of *ewp* from the formula. Note that the formula can only estimate the real baseline price. The real value of the tariff varies from customer to customer, depending on their economic preferences.

Definition 6 (Bottom-line Price). *Let τ be a tariff candidate of AgentUDE17. Then the bottom-line price, denoted as p_{bottom}, must be less than the average rate, which satisfies the condition $p_{bottom} < p_{avg}^{\tau}$. Bottom-line price is obtained through estimating the wholesale market costs.*

After all, the bottom-line price determines the lower boundary of the average tariff price so that AgentUDE17 can safely optimize the tariff parameter set, instead of searching in the full search space.

4 Genetic-Based Optimization

Genetic algorithms (GA) are used in many large optimization problems where the problem cannot be solved easily due to its large search space [6]. In this section, we detail how we applied GA to solve the retail trading problem on the fly.

As noted before, AgentUDE17 has only one active tariff, which is required by the algorithm due to trackability of the changes in the tariff specification. At the beginning of the games, AgentUDE17 always start with an offline data, called bootstrap data *BD*.

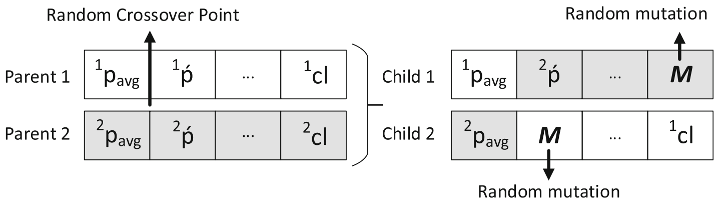


Fig. 1. Crossover and mutation steps in our genetic algorithm.

The bootstrap data initializes the variables that will create the initial tariff τ_{ude}. To do it, we copy individuals and initial tariffs of competitors from BD to the local repository:

$$C_t^1, C_t^2, \ldots, C_t^N \leftarrow BD\left[C_{BD}^1, C_{BD}^2, \ldots, C_{BD}^N\right] \tag{3}$$

$$\tau_1, \tau_2, \ldots, \tau_i \leftarrow BD[\tau_{B_1}, \tau_{B_2}, \ldots, \tau_{B_i}] \tag{4}$$

The individuals are sorted in a way that the healthiest individual is always placed at the first position whereas the weakest one is placed at the last position. Once the repositories are initialized with BD and after every assessment period, we decrease the fitness values of all individuals with a negative constant δ_{bias} to prevent the biased behavior. The biased behavior occurs when an individual has relatively high fitness value among others and its chromosomes are outdated due to a significant change in the economic environment. Therefore, we make sure that a new individual is added to the population, ranked in a top position. We define the bias constant as follows:

$$\delta_{bias} = -\left|f_{C_t^1} / \delta_{bias-cons}\right| \tag{5}$$

where $\delta_{bias-cons}$ is an experimentally determined integer. In addition to the competitors' tariffs (see Formula 2), new published tariffs are added to the tariff repository. Therefore, we calculate the baseline price before and after each assessment period:

$$p_{base} = \min_{i \neq ude} p(\tau_i) \tag{6}$$

where i is the number of active tariffs in the tariff repository. One cycle in GA is called a generation, and it consists of selection, crossover and mutation steps, as shown in Fig. 1. A new child is created just after the assessment period.

In the selection process, we only pick the first two individuals as new parents from the repository, which is called elitism. The reason behind it that tariff specifications can rapidly be outdated due to changing conditions either in the markets or in the retail policies of competitors. In the crossover process, we determine a random point between genes. As illustrated in Fig. 1, the genes on the left side are kept in child 1 and child 2, whereas the ones on the right side are appended to child 2 and child 1, respectively. The crossover process results in two new children. Afterwards, we randomly pick one of the children to apply the mutation process. The new chromosome is denoted as $C_{new} = \left\langle p_{avg}^{\tau_{ude}}, \acute{p}^{\tau_{ude}}, bp^{\tau_{ude}}, ewp^{\tau_{ude}}, cl^{\tau_{ude}} \right\rangle$.

$$C_{new} \leftarrow crossover\left(C_t^1, C_t^2\right) \tag{7}$$

Which is subject to (8) and (9):

$$p_{bottom} \leq p_{base} \tag{8}$$

$$p_{base}\left(1 - \varepsilon_p\right) \leq p_{avg}^{\tau_{ude}} \leq p_{base}\left(1 + \varepsilon_p\right) \tag{9}$$

where ε_p is a mutation constant and it determines the value range in which tariff prices can be defined. Therefore, if condition (9) is not satisfied, then $p_{avg}^{\tau_{ude}} = p_{base}$. Likewise, if condition (8) is not satisfied, then $p_{avg}^{\tau_{ude}} = p_{bottom}$. We apply a random mutation to all genes, subject to the constraints (8) and (9). We randomly define the new values, satisfying the following conditions:

$$p_{avg}^{\prime\tau_{ude}} \in \left[p_{avg}^{\tau_{ude}}\left(1 - \varepsilon_p\right), p_{avg}^{\tau_{ude}}\left(1 + \varepsilon_p\right)\right] \tag{10}$$

$$\grave{p}^{\prime\tau_{ude}} \in \left[\grave{p}^{\tau_{ude}}\left(1 - \varepsilon_p\right), \grave{p}^{\tau_{ude}}\left(1 + \varepsilon_p\right)\right] \tag{11}$$

$$bp^{\prime\tau_{ude}} \in \left\{bp^{\tau_{ude}} + \varepsilon_{bp\&ewp}, bp^{\tau_{ude}} - \varepsilon_{bp\&ewp}\right\} \tag{12}$$

$$ewp^{\prime\tau_{ude}} \in \left\{ewp^{\tau_{ude}} + \varepsilon_{bp\&ewp}, ewp^{\tau_{ude}} - \varepsilon_{bp\&ewp}\right\} \tag{13}$$

$$cl^{\prime\tau_{ude}} \in \left\{cl^{\tau_{ude}} + \varepsilon_{cl}, cl^{\tau_{ude}} - \varepsilon_{cl}\right\} \tag{14}$$

Where mutation constants $\varepsilon_{bp\&ewp}$. and ε_{cl} are used for bonus and penalty payments and contract length, respectively. Tariff prices in (10) and (11) use a multiplier to determine their new values whereas (12), (13) and (14) are defined, by means of incrementing and decrementing the original value. New values constitute the mutated child $C_{new}^{mutated} = \left\langle p_{avg}^{\prime\tau_{ude}}, \grave{p}^{\prime\tau_{ude}}, bp^{\prime\tau_{ude}}, ewp^{\prime\tau_{ude}}, cl^{\prime\tau_{ude}}\right\rangle$.

After all, the new mutated child is transformed into a tariff specification $\tau_{ude} \leftarrow C_{new}^{mutated}$ and submitted to the retail market as of current time slot t. It will remain active during a new assessment period, literally until time slot $t+\delta$. On the other side, the preceding active tariff $\tau_{ude}^{t-\delta}$, that is defined at time slot $t-\delta$ must assessed, using the fitness function. It is added to a position in the population, ranked by its fitness value. The assessment process is skipped at the beginning of games.

As mention before, we use a TOU price scheme in the tariff specifications and the unit consumption value p_{avg} only indicates the average price of the scheme. Therefore, we transform p_{avg} into a detailed TOU price scheme, using the wholesale market costs and grid-wide distribution data. Average market clearing price are denoted as cp_h^{wd} and cp_h^{we} separately for weekdays and weekends, where $h \in \{1, 2, \ldots, 23\}$. Likewise, average hourly distribution is denoted as D_h^{wd} and D_h^{we}. Note that the distribution data reflects the net distributed energy of all brokers, and customer-based production amount is substracted from the total consumption. Then we obtain two vectors, $(\omega_1^{wd}, \omega_2^{wd}, \ldots, \omega_{24}^{wd})$ and $(\omega_1^{we}, \omega_2^{we}, \ldots, \omega_{24}^{we})$, where $\omega_h = cp_h^{wd} D_h^{wd}$. Then we normalize these vectors, satisfying $\sum \omega_{24}^{we} = 24$ and $\sum \omega_{24}^{wd} = 24$. After all, we define the TOU rates for weekdays and weekends as follows:

$$\left\{p_1^{\prime\tau_{ude}}, p_2^{\prime\tau_{ude}}, \ldots, p_R^{\prime\tau_{ude}}\right\} \leftarrow \left\{p_{avg}^{\prime\tau_{ude}} \omega_1, p_2^{\prime\tau_{ude}} \omega_2, \ldots, p_{24}^{\prime\tau_{ude}} \omega_{24}\right\} \tag{15}$$

In a standard tariff of AgentUDE17, we define 48 rates, representing every hour on weekdays and weekends. The flowchart of the algorithm can be summarized in Fig. 2.

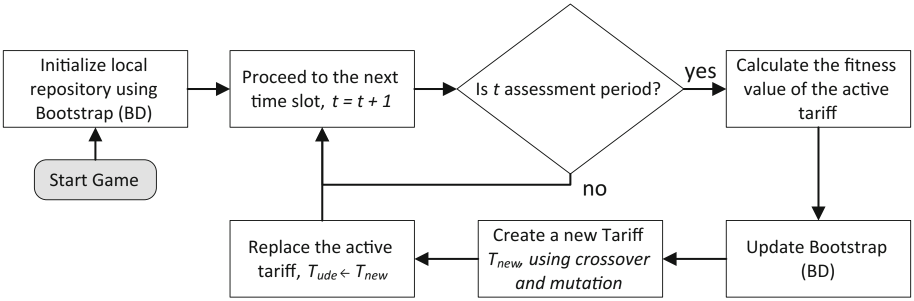


Fig. 2. Flowchart of the proposed algorithm.

5 Results and Discussion

We deployed the presented approaches in our winning broker AgentUDE17 that was extensively challenged in the Power Trading Agent Competition 2017 Final. Therefore, we will analyze the tournament data to validate our approach (proof of concept).

5.1 Tournament Data and Analysis Process

The Power TAC 2017 final games were played in June 2017 immediately after two trial tournaments and a qualification round. Eight brokers competed in 284 games. From those games, 60 were played in 8-player mode whereas 112 games were played in both 5-player and 3-player modes. The brokers in the finals were: AgentUDE17 [8], COLDPower17, CrocodileAgent [1], SPOT [2], VidyutVanika, ewiBroker, fimtac, maxon17 [10]. AgentUDE won the tournament using the methods in this paper.

After a game is successfully completed, a state log file is created that contains all financial and energy transactions. All relevant outputs were processed in MATLAB R2016b and are valid for Power TAC version 1.4.3.

5.2 Results

In this section, we evaluate the performance of the algorithm, based on tournament data. We included all the games from the Power TAC 2017 Final tournament, in which AgentUDE17 is involved. Out of 172 games in total, 60 games were in 8-player, 70 games were in 5-player and 42 games were played as 3-player games.

In the tournament, we optimized the tariff parameters based on a single active tariff. Therefore, our algorithm needs to publish and revoke tariffs after short intervals to measure the fitness value of the tariff. Figure 3 shows the cost of running the algorithm.

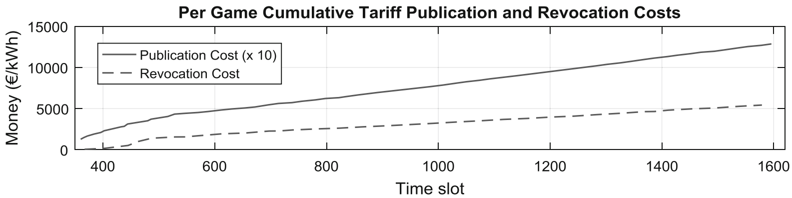


Fig. 3. Per game cumulative sum of tariff publication and revocation fees.

As mentioned earlier, we use a TOU price scheme in the tariffs for a few reasons. Firstly, we reduce the peak-demand charges. Peak-demand charges may be extremely high during the peak hours and may result in a huge loss in the broker's bank account. We use high rates for peak hours and low rates for off-peak hours to solve this problem. Therefore, we encourage our customers to avoid consumptions during peak hours.

As seen in Fig. 4, AgentUDE17 motivated its customers to shift (i.e. valley filling) their loads to the preceding or proceeding time slots. The same figure (right panel) compares the peak-demand charge rates of the brokers.

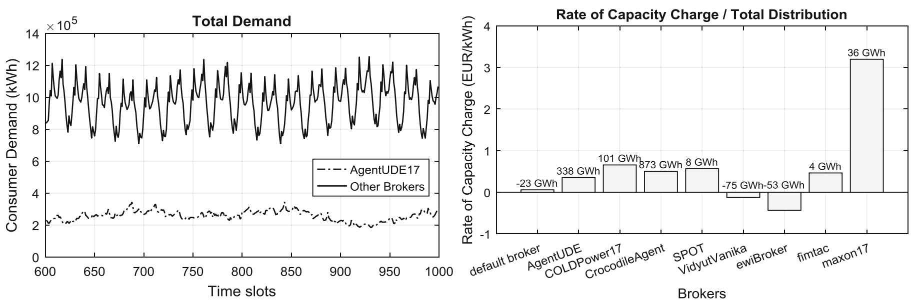


Fig. 4. Left panel indicates the total demand of AgentUDE17 and the total demand of other brokers. A TOU price scheme drives the motivation of customer demand. Right panel shows the rate of total peak-demand charges to total distribution. Annotations refer to total distribution in GWh. Negative distribution refers to local production, which is subject to production tariffs.

Figure 5 illustrates the average cash position of AgentUDE17 in 3-, 5- and 8-player modes. In 5- and 8-player games, AgentUDE17 managed to minimize its peak-demand charges and increase its profit steady. Especially, in 5- and 8-player games, AgentUDE17 could not attract many customers, due to limited profitability space. Adding the tariff publication and revocation fee (see Fig. 3) on top of average cash positions, AgentUDE17 managed to stay on the profitable side.

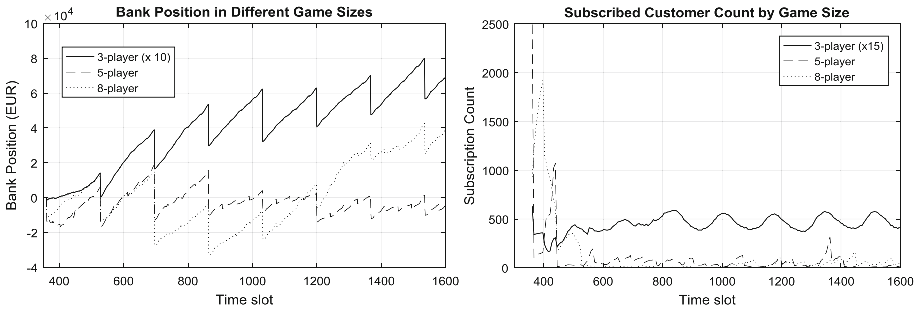


Fig. 5. Left panel indicates the average cash position of AgentUDE17 in different game sizes. Sharp declines in the bank position refer to peak-demand charges, paid once a week (168 time slots). Right panel shows the number of subscribed customers to the AgentUDE17 tariffs. The count for the 3-player games is 15 times higher than the indicated.

Figure 6 illustrates the mean and standard deviation of tariff rates in AgentUDE17 tariffs (left panel) and average tariff price of the lowest 10 tariffs of competitors (right panel). On the left panel, the standard deviation and the mean are inversely proportional to the game size. The reason is that 8-player games are repeated with the same brokers.

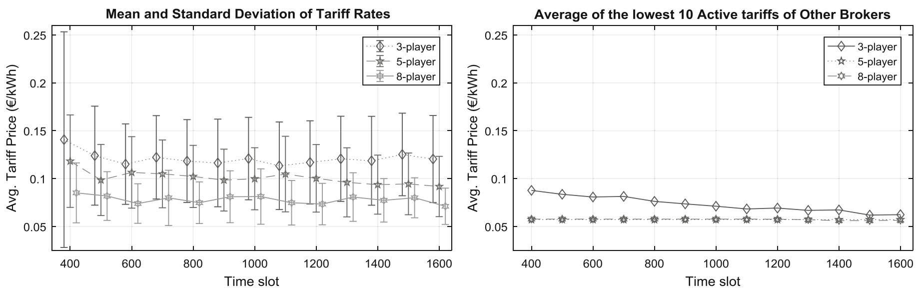


Fig. 6. Left panel illustrates the mean and standard deviation of tariff rates in short intervals. Right panel indicates the average of the lowest 10 active consumption tariffs of the competitors.

Figure 7 illustrates the mean and standard deviation of periodic payment (left panel) and the cumulative sum of periodic payment by game size (right panel). In 5- and 8-player games, we see a steady increase in the value of the periodic payment. The reason behind it that AgentUDE17 attracts few customers in 5- and 8-player games due to the tough competition. The level of the competition is relatively lower in 3-player games. Especially, in 8-player games, the periodic payment is optimized in a larger search space. Due to the number of customer count in 3-player games, it provided the highest earnings, as shown in the right panel.

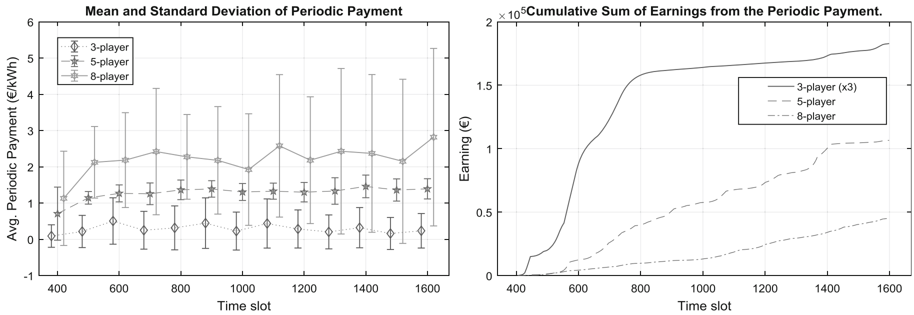


Fig. 7. Left panel shows the mean and standard deviation of periodic payments. Right panel illustrates the per game earnings of AgentUDE17 from the periodic payment.

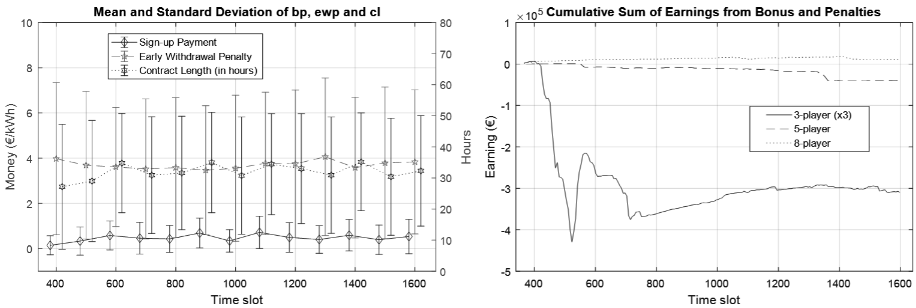


Fig. 8. Left panel illustrates the mean and standard deviation of tariff fees and contract length. Right panel shows the cumulative earnings from the tariff bonus and penalties.

Figure 8 illustrates the mean and standard deviation for the bonus payment, early withdrawal penalty, and contract length as well as the cumulative sum of earnings, collected from these payments. As seen in the right panel, AgentUDE17 paid the most amount of money in 3-player games. AgentUDE17 gained some profit in 8-player games.

6 Conclusion and Future Work

In this paper, we presented a genetic algorithm to optimize the tariff parameters of an electricity tariff on the fly. The broker always starts with an offline data, called bootstrap database, which is picked from a decision tree structure. In the bootstrap data, we basically keep the most recent knowledge of the broker as well as competitive tariffs of other brokers. During a game, the algorithm periodically produces a set of parameters. The broker modifies the active tariff, replacing its parameters with the new ones. However, the algorithm selects the parameter values after many evolutionary processes so that it maximizes the profitability in long term.

We employed the algorithm in our winning broker agent AgentUDE17. We investigated the Power TAC 2017 Final tournament, analyzing its game logs files. Results show that AgentUDE17 succeeded to be on the profitable side, despite a tough competition, especially in 8-player games. Besides, it managed to minimize its peak-demand charges by means of using TOU price schemes.

Broker agents trade in a highly dynamic environment. This means that they deal with a huge number of variables on the fly. Many of them fail in terms of finding the global optimum, despite using advanced machine learning algorithms. In AgentUDE17, we used a genetic algorithm (GA) in the retail market and realized that GA fits extremely well to such dynamic environments. As a part of this project, we plan to implement an evolutionary bidding mechanism on the wholesale market since we think that autonomous traders (not limited to power traders) need an advanced self-evaluation structure to fully explore the search space and adapt themselves to new market conditions.

References

1. Babic, J., Podobnik, V.: Adaptive bidding for electricity wholesale markets in a smart grid. Presented at the AAMAS Workshop on Agent-Mediated Electronic Commerce and Trading Agents Design and Analysis (AMEC/TADA 2014) (2014)
2. Chowdhury, M.M.P., Folk, R.Y., Fioretto, F., Kiekintveld, C., Yeoh, W.: Investigation of learning strategies for the SPOT broker in power TAC. In: Ceppi, S., David, E., Hajaj, C., Robu, V., Vetsikas, Ioannis A. (eds.) AMEC/TADA 2015-2016. LNBIP, vol. 271, pp. 96–111. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54229-4_7
3. Farhangi, H.: The path of the smart grid. IEEE Power Ener. Mag. **8**(1), 18–28 (2010)
4. Hoogland, J., La Poutré, H.: An effective broker for the power TAC 2014. In: Ceppi, S., David, E., Hajaj, C., Robu, V., Vetsikas, Ioannis A. (eds.) AMEC/TADA 2015-2016. LNBIP, vol. 271, pp. 66–80. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54229-4_5
5. Ketter, W., et al.: Power TAC: a competitive economic simulation of the smart grid. Ener. Econ. **39**, 262–270 (2013)
6. Mitchell, M.: An Introduction to Genetic Algorithms. MIT press, Cambridge (1998)
7. Morris, C., Pehnt, M.: Energy Transition: The German Energiewende. Heinrich Böll Stiftung, Berlin (2012)
8. Ozdemir, S., Unland, R.: The strategy and architecture of a winner broker in a renowned agent-based smart grid competition. Web Intelligence **15**(2), 165–183 (2017). IOS Press
9. Rúbio, T.R.P.M., Queiroz, J., Cardoso, H.L., Rocha, A.P., Oliveira, E.: TugaTAC broker: a fuzzy logic adaptive reasoning agent for energy trading. In: Rovatsos, M., Vouros, G., Julian, V. (eds.) EUMAS/AT -2015. LNCS (LNAI), vol. 9571, pp. 188–202. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-33509-4_16
10. Urban, T., Conen, W.: Maxon16: A Successful Power TAC Broker (2016)

11. Urieli, D., Stone, P.: TacTex'13: a champion adaptive power trading agent. In: Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems, pp. 1447–1448. International Foundation for Autonomous Agents and Multiagent Systems (2014)
12. Wang, X., et al.: A hybrid-learning based broker model for strategic power trading in smart grid markets. Know.-Based Syst. **119**, 142–151 (2017)

A First Step Towards a General-Purpose Distributed Cyberdefense System

Aarón Rodríguez and Luis Castillo(✉)

Department of Computer Science and Artificial Intelligence,
University of Granada, Granada, Spain
rodriguez_bueno@hotmail.com, L.Castillo@decsai.ugr.es

Abstract. Most part of the activity in cybersecurity consists of perceiving the state of a distributed environment composed of a network of interconnected computers or devices, taking care of every individual perception of a device but also observing the whole picture of the network. When needed, there is a reaction by executing actions aimed at producing some preventive, defensive or restoring effect in that environment in order to keep it in a safe state. This is, nearly by definition, an ideal area of application for multi-agent systems. This paper presents the first steps given to build a general-purpose multi-agent system for the cyberdefense of a network of computers. It also shows a preliminary experiment to measure the burden of implementing this distributed defense mechanism in terms of CPU and memory overload.

Keywords: Cyberdefense · Distributed problem solving
Embodied agents and autonomous systems

1 Introduction and Motivation

Cybersecurity problems are one of the three top risks as perceived by general society [11] and also a very promising area of application for multi-agent technology [17]. The reasons for this interest lie at the complementarity of the requirements of many cybersecurity problems and the technical features offered by multi-agent systems. The arrival of the Internet of Things has produced a massive presence of interconnected devices of different nature, but the security issues of these devices is often neglected [9]. It is becoming increasingly clear that attacks occur more rapidly and more pervasive than ever before[1], making it necessary to rely on automated and cognitive technologies [2,9] able to give an appropriate and timely response to mitigate those incidents and restore the appropriate level of service as soon as possible. Even more, at private-corporate or public-institutional levels, the detection and reaction to security incidents

[1] DARPA Cyber Grand Challenge (2016) http://archive.darpa.mil/cybergrandchallenge/.

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 237–247, 2018.
https://doi.org/10.1007/978-3-319-94580-4_19

need to be coordinated and homogeneous, taking into account the multiplicity of hardware and software platforms of the deployed IT Systems which might range from classical computers to advanced devices like connected cars or drones.

Given the inherent distributed nature of multi-agent systems, their interconnecting and proactive capabilities [19] and also its multi-platform deployments [3] make this technology very suitable to provide the technological scaffolding for building cyberdefense systems into heterogeneous IT Systems. Multi-agent systems and platforms offer an integrated framework for deploying homogeneous agents, written in Java for instance [18], over heterogeneous infrastructures, provide the communication channels amongst them and foster the auto-organization, cognitive and problem-solving capabilities [16] needed to succeed in surveillance and reaction goals. Even more, the increasing implementation of software defined networks [14], in which network assets like routers or switches may be virtualized by using a regular computer able to run complex applications, facilitates the introduction of complex software artifacts, like the agents presented in this paper, deep in the network infrastructure, with powerful problem-solving and cognitive capabilities.

This paper presents ongoing research in this direction, introducing CID [13], which stands for the Spanish acronym of Distributed Intelligent Cyberdefense, a distributed cyberdefense system implemented as a FIPA-compliant multi-agent system over a distributed IT infrastructure able to monitor certain cyber-threats, provide a local response when needed and coordinate a global coordinated response when the vector attack might come from different sources towards multiple target nodes in the defended network, like a botnet attack [1]. Although the approach only covers one type of cyberthreat yet, it is easily extendable to cover many other threats and the paper is complemented with an empirical study on the scalability of the multi-agent system, in terms of CPU and memory usage, when the number of threats and the number of agents deployed increases from a few of them to hundred agents and threats, by using the Magentix2 [18] agent platform.

The structure of the paper is as follows. Section 2 reviews related work and put the paper under an appropriate perspective. Section 3 introduces the types of agents, their relationship and the type of threats covered in the paper. Later, Sect. 4 introduces an experimental study designed to quantify the computational burden of implementing the approach and Sect. 5 draws some conclusions and further lines of work.

2 Multi-agent Cyber-Security Approaches

There is a vast amount of applications of multi-agent technology to the very numerous and different aspects of network security showing that cybersecurity is a hot topic [16,17]. Most of them are focused on distributed and collaborative monitoring, simulation and identification of anomalies in networks through different domains. In [15] authors describe a distributed intrusion detection system composed of two layers of agents to detect anomalies in industrial SCADA networks. Work presented in [6,8] shows an agent-based modeling to simulate complex vector attacks which are used to build the cyberdefense of a network. Specially remarkable is the first work [8] in which the protected network belongs to a modern commercial passenger aircraft, what highlights that the domain of cyberdefense is not limited to classical networked computers. Also related to this, in [10] authors introduce a military application in which a set of defender agents collaborate and share information in order to build a mental representation of the attacker based on a game-theoretic approach.

The use of multi-agent technology to orchestrate a defense against certain attacks is shown in [12] where a distributed agent-based cooperative framework is proposed for the protection of existing systems in modern power grids. The agents deployed over a physical industrial asset are able to monitor low-level electric signals and process them to identify a set of cyberthreats and react by sending blocking electric signals. Also in [7] a multi-agent system is used to detect a threat over mobile communication networks and to restore the compromised connectivity of the network. This is also the topic of this paper but from a general-purpose perspective.

3 CID Architecture

CID is designed to be a multi-purpose, easily scalable FIPA-compliant multi-agent architecture to detect and mitigate cybersecurity incidents in a certain network and the initial architecture introduced in this paper is shown in Fig. 1. It has been implemented in Magentix agent programming platform [18] and it is written in Java[2].

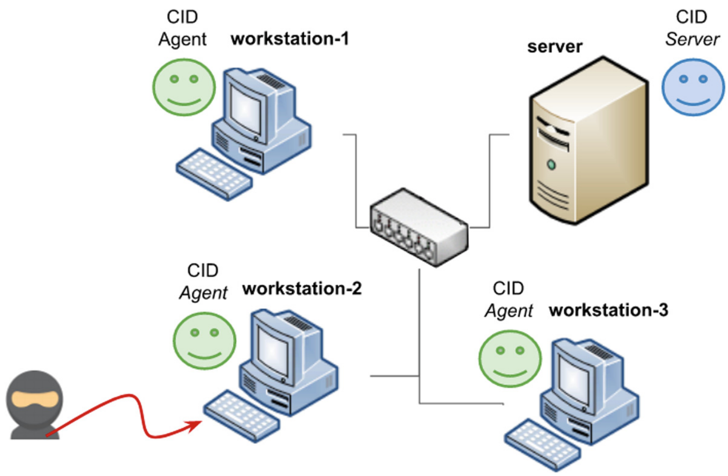


Fig. 1. Initial distributed architecture of CID

[2] The complete source code of CID is availabe on GitHub https://github.com/HaruGaren/CID.

There is a CID Agent deployed at every workstation in the network which monitors its host system looking for evidences of a possible threat in system logs and activity logs. Whenever a threat is found, the corresponding agent tries to mitigate it by executing certain system scripts and restore the level of service as soon as possible. All threats are communicated to a server running a CID Server Agent[3]. This CID Server Agent correlates the information coming from the different CID Agents connected to the network and, in the case that it detects a coordinated attack towards several nodes, like for example coming from a botnet, it informs back all the CID Agents and spread the required information to reach a coordinated response of the whole defense.

3.1 Threats, Detection and Response

CID architecture is still evolving but it is completely functional to date. It only covers yet, as a sample incident, one type of threat, a ssh brute-force attack, that is, repeated attempts to access ssh by guessing the password of a known user using brute force, or trying to guess potentially vulnerable authorized users, and the only supported operating system is Linux. But it does covers the whole life-cycle of an incident management system by means of the capabilities of every CID Agent deployed in the network: they monitor their system, exchange information with the server, mitigate the threat and restore the initial level of service.

This brute-force attack can be early detected by the CID agent just by looking at the log of errors of the authorization system (in Linux) in `/var/log/auth.log` and looking for a pattern which means repeated authorization failures from the same IP (see Fig. 2a). In this case, when the number of failures exceeds a given threshold, the CID agent blocks the corresponding IP by adding it to the `iptables` register of the operating system.

This reaction of the CID agent (see Fig. 2b) prohibits the suspicious IP to connect to the host but since this might also be a legitimate action, due to a forgotten password, for example, the CID agent re-allows again the banned IP to connect after a given lapse of time (Fig. 2c).

This is a toy example used to illustrate the life-cycle of the CID agent to detect and mitigate a given threat and also to restore the level of service, and it is included here only for this purpose. More complex threats can be added to the defense capabilities of CID just by looking for the required patterns in the corresponding logs of the operating system and reacting accordingly, either with a common single agent or by letting each threat to be handled by a specialized agent. It is also possible to cover other operating systems like Windows or MacOS as long as they support Java and Magentix libraries and agents are rewritten to look for operating-system specific logs, what is out of the scope of this paper.

[3] Technically this server is also used to host the agent platform, but both, the platform and the CID Server Agent might be up and running on any computer of the network.

```
Jan 26 10:07:30 HOST sshd[25579]: pam_unix(sshd:auth): authentication failure; logname= uid=0
   euid=0 tty=ssh ruser= rhost=any.computer.net  user=peter
Jan 26 10:07:31 HOST sshd[25579]: Failed password for peter from 172.20.2.125 port 59749 ssh2
Jan 26 10:07:37 HOST sshd[25579]: Failed password for peter from 172.20.2.125 port 59749 ssh2
```

(a)

```
# iptables -A INPUT -s 172.20.2.125 -j DROP
```

(b)

```
# iptables -D INPUT -s 172.20.2.125 -j DROP
```

(c)

Fig. 2. (a) Pattern found in `/var/log/auth.log` showing two consecutive failed trials to access user `peter` from IP `172.20.2.125` due to wrong password. (b) Call to `iptables` to block the suspicious IP (c) Call to `iptables` to unblock the previously banned IP

3.2 Coordinated Response to Multiple Vector Attacks

All the threats detected by any CID agent are also communicated via FIPA Agent Communication Language[4] to the Server Agent, who records them and, in the case of detecting the same IP, or IPs coming from the same subnet, threatening more than one CID agent, it is spread to the whole set of agents in order to prevent possible coordinated attacks.

3.3 Resilient Functional Units

CID Agent and CID Server are regular Magentix agents [18] written in Java. They record their full activity on disk, so their normal operation can be restored in the case of shutdown of the network or the host computer, just by resuming their activity exactly in the point of surveillance or reaction in which they were just before they broke. The abstract operation of these agents is described in Figs. 3 and 4 respectively and it is independent of the agent platform and the running operating system.

Every CID Agent (Fig. 3) starts up and it checks whether the host operating system is a compatible one and, if positive, it connects to the CID Server Agent to register its presence in the system and to subscribe to other potential threats informed by the CID Server Agent. Immediately after, it starts parsing the log file for matching patterns of threats. If a threat is found, the CID Agent executes the command in Fig. 2b, so that the detected IP or IPs are banned to the system and the CID Server is informed. If a CID Agent receives more IPs from the CID Server Agent, coming from attacks to other nodes of the network the CID Server Agent is aware of, they are also banned and the cycle starts again. Banned IPs are re-allowed to connect to the system after a certain amount of time.

[4] FIPA ACL Specification http://www.fipa.org/repository/aclspecs.html.

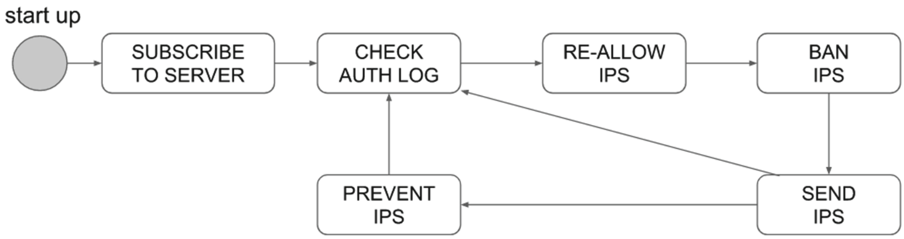


Fig. 3. FSA of the CID agent

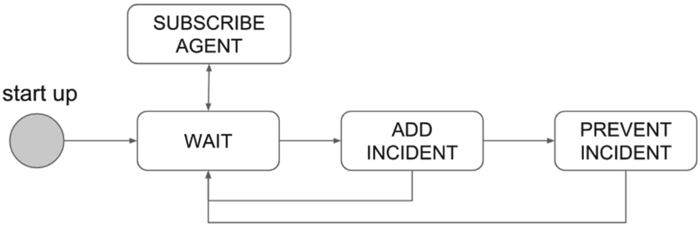


Fig. 4. FSA of the CID server

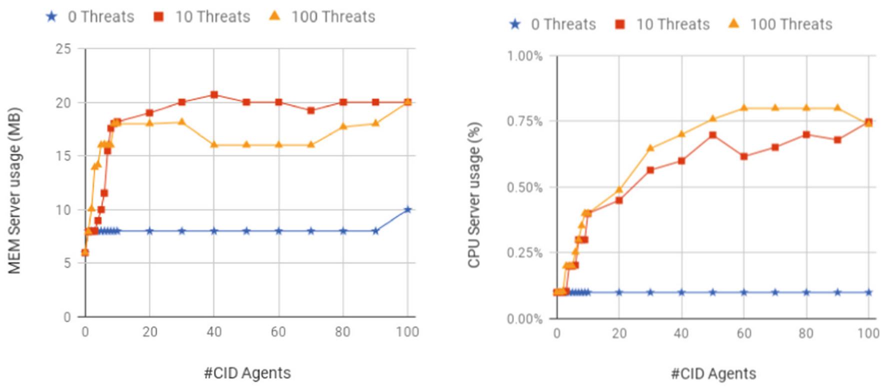
There is a single CID Server Agent running in the whole architecture (Fig. 4), indeed it must be the first launched agent, and it maintains a list of subscribed agents in the network, that is, the set of agents deployed to protect the infrastructure. As soon as it receives a warning message from any of them, it adds it to a list of active threats and, in the case of more than one agent reports the same incident, the respective IP is broadcasted among all the subscribed agents to prevent potential coordinated attacks. CID Agents are specific to the operating system on which they run, since they execute system's scripts and monitor system logs at operative level, but the CID Server is a completely autonomous one since it only centralizes and coordinate the information coming from the deployed CID Agents.

Next section shows the results of several experiments run to test the performance of this architecture by scaling up the number of detected threats and the number of deployed agents as a first step to validate the scalability of the CID Architecture to cover a higher number of different threats and/or larger networks.

4 Experiments

Having a single idle agent monitoring a single log file means nothing in terms of CPU and memory usage, either for a workstation (CID Agent) or for a server (CID Server Agent). However realistic cases might well involve many deployed agents and many monitored threats. Since, to date, CID only supports a single type of threat, the following experiment has been designed. Let CID run over an increasing number of deployed agents in the network (from 0 to 100) in different scenarios ranging in the number of simultaneous incidents from 0 threats (idle agent) to 100 threats, all of them of the same type, and let us measure the use of CPU and memory in each case, both in the server and in workstations.

Figure 5(a) shows the impact on memory usage in the server side and it shows a quite stable performance (between 15 and 20 MB used) even in extreme cases with 100 agents and 100 simultaneous incidents each. Figure 5(b) shows the impact on CPU usage which does not exceed a 1% of overload, even in extreme cases. These charts also show that the impact of the communication effort between agents is also quite affordable despite all detected incidents are communicated to the CID Server (the content of all messages is sent in JSON[5] format with only a few tens of bytes in length each, but hey can be compressed in the case of longer strings in order to avoid communication overhead).



(a) (b)

Fig. 5. Experimental results showing the impact of the memory usage (left) and the CPU usage (right) of the CID approach in the server for a number of CID Agents ranging from 0 to 100 at three different scenarios with 0, 10 and 100 threats detected per each agent. Server = Intel XEON 2GB, Linux Ubuntu 12.0.4, 1 Gbit ehternet connection

Figure 6 repeats a similar experiment on the workstation side but, in this case all the deployed agents, from 0 to 10, run on the same workstation at the same time in a trial to model more heterogeneous CID architectures monitoring more complex and varied incidents.

Figure 6(a) shows an almost linear impact in memory usage, although it is not critic, for a number of agents less than 10. It is also quite affordable even with 100 simultaneous incidents. Figure 6(b) shows the impact in CPU usage which, in the worst case, that is, 10 agents monitoring 100 simultaneous incidents each of them, the CPU usage scales up to a 10% of overload. This is not excessive

[5] JavaScript Object Notation https://en.wikipedia.org/wiki/JSON.

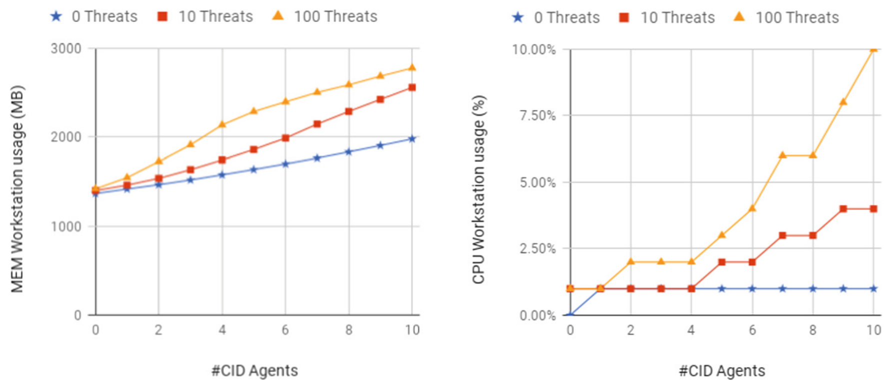
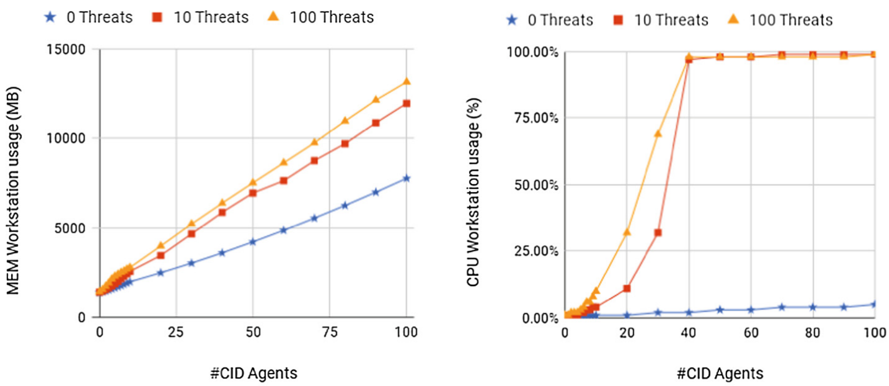


Fig. 6. Experimental results showing the impact of the memory usage (left) and the CPU usage (right) of the CID approach in the workstation for a number of CID Agents ranging from 0 to 10 at three different scenarios with 0, 10 and 100 threats detected per each agent, all of them running in the same workstation. Workstation = Intel Core i7 16 GB, Linux Ubuntu 16.0.4, 1 Gbit ehternet connection



(a) (b)

Fig. 7. Extension of the experiment in Fig. 6 for a extreme case of having up to 100 CID Agents running in the same computer monitoring a varying amount of incidents

taking into account that any distributed defense artifact will have an obvious impact on CPU, but it points to what seems to be an exponential growth of the CPU usage. In order to unveil what would happen with an extreme number of agents running on the same workstation, a new experiment is carried out with the same configuration, but with an asymptotic range of up to 100 CID Agents running on the same workstation.

Figure 7(a) still shows a linear growth in the memory usage which dramatically scales up to 80% of available memory in the most extreme (rather unrealistic, indeed) case. However Fig. 7(b) shows that the CPU collapses when the number of agents exceeds of 30 agents, independently of the number of incidents (neglecting the idle case with 0 threats).

This last experiment, although in a simulated environment just make clear the obvious: there is a number of agents which should not be exceeded not to collapse the regular operation of the workstation. The exact number of agents that lead to a collapse of the workstation depends on the regular CPU consumption of the workstation and the variety and nature of the threats being monitored, and even the operating system, but in the ideal case of Fig. 7, with no other process running on the same workstation than multiple instances of CID Agents, having 40 CID Agents collapses the workstation.

5 Conclusions and Future Work

CID aims to be a general-purpose multi-agent framework to detect potential threats over a network of computers, react to mitigate them and restore the appropriate level of service. This complete life-cycle has been shown regarding a single, simple threat: bruteforcing the password of a known user account. This threat, as many others, may be detected by looking for certain patterns in the logs of the operating system (up to now the only supported OS is Linux). Its mitigation and recovery can also be achieved by letting agents execute certain operating-system commands and scripts. All these perception-reaction activities are executed from within every CID Agent and communicated to the CID Server Agent in order to detect potential coordinated attacks like a botnet. Every agent in CID may be easily deployed at any node of the network[6] since, at agent's start up, it has local reaction capabilities and it connects to the CID Server to achieve a global reaction too.

In addition to this, the burden in terms of CPU and memory usage has also been sketched in a series of experiments showing that CID may scale up very well but system performance might be dramatically affected when the number of simultaneous agents running on the same computer is high (more than 40 in the running experiments). Whether this asymptotic number of agents is a realistic approximation or not would require advancing the capabilities of the agents to cover more threats, something that can be easily achieved whenever the threats leave a footprint in system logs and the reaction might be arranged by executing sequences of shell scripts and commands. This is the main subject for further study: reinforce the defense capabilities and keeping the resource consumption bounded.

Honestly there is an important weakness of the approach that must be noted. It is due to the multi-agent platform used, which is also general to most, if not all, the multi-agent platforms. Any instance of the CID Agent and CID Server Agent may be deployed on any node of the network, but only one node of the network, which could be any of the computers of the network, contains the low level communication broker module of the agent platform. This communication

[6] Every agent is simply a JAR file with its associated initialization file encoded in JSON.

broker is the one that receives ACL messages from the sender agent, and routes them to the corresponding receiving agents. This might be a potential flaw of the architecture since this node might be blocked or under attack and, although the local reaction of the CID Agents would not be affected, the communication and coordination among agents would be affected or even fully blocked.

Another line for future research would be enhancing the capabilities of CID Server agent with more powerful cognitive capabilities. It is the agent with the global view of the situation and it may use Artificial Intelligence Planning and Scheduling technologies to autonomously compose complex reaction plans to face an emergency situation, as it is shown in [4,5]. This reaction plan might be distributed among enhanced versions of CID Agents to execute the plan and to adapt it to the specific operating system they are running on.

Acknowledgements. This work has been partly funded by TIN2015-71618-R (MINECO/FEDER).

References

1. Abu Rajab, M., Zarfoss, J., Monrose, F., Terzis, A.: A multifaceted approach to understanding the botnet phenomenon. In: Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, pp. 41–52. ACM (2006)
2. Avgerinos, T., Cha, S.K., Rebert, A., Schwartz, E.J., Woo, M., Brumley, D.: Automatic exploit generation. Commun. ACM **57**(2), 74–84 (2014)
3. Dastani, M.: A survey of multi-agent programming languages and frameworks. In: Shehory, O., Sturm, A. (eds.) Agent-Oriented Software Engineering, pp. 213–233. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54432-3_11
4. de la Asunción, M., Castillo, L., Fdez-Olivares, J., García-Pérez, O., González, A., Palao, F.: SIADEX: an interactive artificial intelligence planner for decision support and training in forest fire fighting. Artif. Intell. Commun. **18**(4) (2005)
5. Fdez-Olivares, J., Castillo, L., García-Pérez, O., Palao, F.: Bringing users and planning technology together. Experiences in SIADEX. In: Sixteenth International Conference on Automated Planning and Scheduling, ICAPS (2006). Awarded as the Best Application Paper of this edition
6. Kotenko, I.: Multi-agent modelling and simulation of cyber-attacks and cyber-defense for homeland security. In: 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2007, pp. 614–619. IEEE (2007)
7. Magán-Carrión, R., Camacho-Páez, J., García-Teodoro, P.: A multiagent self-healing system against security incidents in MANETs. In: Corchado, J.M., et al. (eds.) PAAMS 2014. CCIS, vol. 430, pp. 321–332. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07767-3_29
8. Mead, J., Vasatka, J.E., Craig, J.A.: Cybersecurity system with differentiated capacity to deal with complex cyber attacks. US Patent App. 14/872,698 6 Apr 2017
9. National Institute of Standards and NIST Technology: Framework for improving critical infrastructure cybersecurity v 1.1 draft 2, December 2017
10. Oltramari, A., Ben-Asher, N., Cranor, L., Bauer, L., Christin, N.: General requirements of a hybrid-modeling framework for cyber security. In: 2014 IEEE Military Communications Conference (MILCOM), pp. 129–135. IEEE (2014)

11. Palmer, D.: Cyber-attacks are a top three risk to society, alongside natural disaster and extreme weather. ZDNet, January 2018. http://www.zdnet.com/article/cyber-attacks-are-a-top-three-risk-to-society-alongside-natural-disaster-and-extreme-weather/
12. Rahman, M.S., Mahmud, M.A., Oo, A.M.T., Pota, H.R.: Multi-agent approach for enhancing security of protection schemes in cyber-physical energy systems. IEEE Trans. Industr. Inform. **13**(2), 436–447 (2017)
13. Rodríguez, A.: Cid. ciberdefensa inteligente distribuida (distributed intelligent cyberdefense). M.Sc. thesis, University of Granada (2017)
14. Shin, M.-K., Nam, K.-H., Kim, H.-J.: Software-defined networking (SDN): a reference architecture and open APIs. In: 2012 International Conference on ICT Convergence (ICTC), pp. 360–361. IEEE (2012)
15. Shosha, A.F., Gladyshev, P., Wu, S.-S., Liu, C.-C.: Detecting cyber intrusions in SCADA networks using multi-agent collaboration. In: 2011 16th International Conference on Intelligent System Application to Power Systems (ISAP), pp. 1–7. IEEE (2011)
16. Singh, M.P.: Cybersecurity as an application domain for multiagent systems. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, pp. 1207–1212. International Foundation for Autonomous Agents and Multiagent Systems (2015)
17. Such, J.M., Criado, N., Vercouter, L., Rehak, M.: Intelligent cybersecurity agents [guest editors' introduction]. IEEE Intell. Syst. **31**(5), 3–7 (2016)
18. Such, J.M., GarcíA-Fornes, A., Espinosa, A.N., Bellver, J.: Magentix2: a privacy-enhancing agent platform. Eng. Appl. Artif. Intell. **26**(1), 96–109 (2013)
19. Wooldridge, M.: An Introduction to Multiagent Systems. Wiley, Hoboken (2009)

Check for updates

A Network-Oriented Adaptive Agent Model for Learning Regulation of a Highly Sensitive Person's Response

Linh Tran, Jan Treur(✉), and Denice J. Tuinhof

Behavioural Informatics Group, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
trplinh@gmail.com, j.treur@vu.nl, denicetuinhof@gmail.com

Abstract. Inspired by the work of Elaine Aron, in this paper a human-like adaptive computational agent model of the internal processes of a highly sensitive person (HSP) is presented. This agent model was used to get a better understanding of what goes wrong in these internal processes once this person gets upset. A scenario is addressed where a highly sensitive person will get upset by an external stimulus and will not be able to calm down by herself. Yet in a social context the interaction with a second person (without high sensitivity) will calm the HSP down, thus contributing to regulation. To obtain an adaptive model a Hebbian learning connection was integrated. During interaction with a second person this Hebbian learning link will become stronger, which makes it possible for a HSP to become independent after some time and be able to regulate upsetting external stimuli all by herself.

Keywords: Highly sensitive person · Hebbian learning
Sensory processing sensitivity

1 Introduction

According to Jagiellowicz *et al*. [1] one fifth of the population has high Sensory-Processing Sensitivity also called (HSP); see also [2, 4, 6, 8, 9]. All these people would profit from more knowledge on their disorder. So, as a lot of people suffer from HSP and still a lot remains unknown, there is a demand for more insight. Therefore, work in this research area is challenging but interesting and would be a good contribution to society. If the internal processes of HSP can be modelled computationally, our understanding of the underlying processes will grow. This enlarged knowledge will support us in the interaction and therapy of highly sensitive persons.

To obtain a human-like computational adaptive agent model, a couple of design decisions were made. First of all, a woman was chosen to be our sensitive person. After reading [5, 7] it was more logical to pick a woman because they are more extreme on the high sensitive person scale, which also includes the reaction on internal and external stimuli. In addition, a negative stimulus was chosen with visual elements (e.g., a flashing light), so that the person may avoid the stimulus by changing her gaze, which

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 248–261, 2018.
https://doi.org/10.1007/978-3-319-94580-4_20

is observable by other persons. Furthermore, a learning aspect was incorporated making the model adaptive. This will give insights in how to make a HSP more independent in upsetting situations. More insight in the learning of a HSP can contribute to therapy for HSP, for example, via a virtual training environment using the model during the education of therapists.

To obtain the human-like agent model, it was decided to use a Network-Oriented Modeling approach as described in [14]. This Network-Oriented approach to agent modeling can be viewed as standing on the one hand in the causal modeling tradition in AI (e.g., [16–18]), and on the other hand in the perspective on mental states and their causal relations in Philosophy of Mind (e.g., [14]). It adds dynamics to causal relations by using an additional temporal dimension, and describes a causal model as a temporal-causal network model. In the temporal-causal network models used, states change over time due to the causal impacts they have on each other, but also the causal relations can change, thus enabling an adaptive network model. In this way learning can be incorporated; for example, Hebbian learning is incorporated by modeling that a connection between two states becomes stronger when both states are active simultaneously.

In the paper in Sect. 2 some background knowledge from Neuroscience is discussed. Next, in Sect. 3 the adaptive agent model is introduced. In Sect. 4 some example simulations are presented. Section 5 discusses the role of requirements and parameter tuning, and Sect. 6 discusses the Mathematical Analysis performed to verify whether the implemented model does what is expected. Finally, Sect. 7 is a discussion.

2 Background Knowledge

This section briefly discusses background knowledge as found in literature such as [1, 5, 7, 9, 10]. A highly sensitive person, or a person with Enhanced Sensory-Processing Sensitivity, is a person which is characterized as being really sensitive to external and internal stimuli, intense emotions and with a preference for deep processing of information [1, 2]. In this research the focus is on the sensitivity for external stimuli [4]. Intense stimuli that may upset a HSP are bright lights, strong smells, coarse fabrics or sirens [4]. Not only in expressions are HSP different from non-HSP. HSP process sensory information much more deeply due to biological differences in their nervous system [7, 8]. These differences and the level of high sensitivity can be measured on a scale. In 1997 Aron and Aron developed a highly sensitive person scale [9]. The HSP scale is a measure of sensory-processing sensitivity, which is conceptualized as involving both high levels of sensitivity to subtle stimuli and being easily over-aroused by external stimuli [10]. This research concludes that men are more moderate on this scale then woman. Which makes in the scenario addressed below a woman more logical as the HSP.

Hebbian learning is a learning mechanism in the (human) brain that suggests that *'units that fire together, wire together'*. According to Hebb the 'efficiency' of a given neuron, in contributing to the firing of another, could increase as that cell is repeatedly involved in the activation of the second [11]. Therefore, neurons that have correlated firing will strengthen their mutual interaction and increase the strength of synapses they form with each other. Certainly, these mutual interactions of neurons do not arise spontaneously. It could take days, weeks or even years for a Hebbian-like mechanism to provide a strong connection. This idea is consistent with the neural mechanisms of long-term potentiation (LTP) and long-term depression (LTD) [12]. LTP increases synaptic efficacy, while LTD weakens the synaptic efficacy. Therefore, LTP and LTD influence the efficacy of synapses or junctions between neurons. Subsequently, the extent to which activity in a sending neuron leads to depolarization of a receiving neuron is influenced as well. While these are long-term processes, it is suggested that Hebbian learning models are highly relevant for investigating development. Moreover, Hebbian learning models can account for a wide range of behaviors and changes during development [12]. Thus, a Hebbian learning model is an eligible addition to the model of a highly sensitive person changing her behaviour over time as addressed below.

The scenario addressed is as follows. After a siren (existing of both light and sound) goes off in the room, this immediately has an effect on Arnie. Arnie is a highly sensitive female and does not like loud noises and bright lights. When Arnie sees the lights, her body language adjusts to the presence of this, in addition she tries to cope with the siren by trying to avoid it with her gaze and speaking her discomfort out loud. Therefore, she communicates this to her friend Bert (a person with normal sensitivity) that the presence of the siren makes her feel uncomfortable. Bert also senses that Arnie's body state is uncomfortable. In addition, Bert notices that Arnie is avoiding the siren with her gaze. These three inputs make Bert want to undertake action and comfort his friend. Bert tries to regulate the situation and prepares some comforting words. He expresses these comforting words to Arnie. Because Arnie avoided the siren by adjusting her gaze, her body is already a little bit less stressed. When her friend Bert supports her with some comforting words, her body relaxes. Arnie needed Bert to comfort her because her internal regulation process is not strong enough to regulate her own internal and body state. However, due to Bert's comforting words, Arnie is learning how to comfort herself in the same way. Through Hebbian learning, Arnie is learning to cope with an upsetting external stimulus on her own a bit better.

3 The Network-Oriented Adaptive Agent Model

First the Network-Oriented Modeling approach used to model the adaptive agent is briefly explained. As discussed in detail in Chap. 2 of [14], this approach is based on temporal-causal network models which can be represented at two levels: by a conceptual representation and by a numerical representation. These model representations can be used not only to display graphical network pictures, but also for numerical simulation. Furthermore, they can be analyzed mathematically and validated by comparing their simulation results to empirical data. They usually include a number of parameters for domain, person, or social context-specific characteristics. A conceptual representation of a temporal-causal network model in the first place involves representing in a declarative manner states and connections between them that represent (causal) impacts of states on each other, as assumed to hold for the application domain addressed. The states are assumed to have (activation) levels that vary over time. In reality, not all causal relations are equally strong, so some notion of *strength of a* *connection* is used. Furthermore, when more than one causal relation affects a state, some way to *aggregate multiple causal impacts* on a state is used. Moreover, a notion of *speed of change* of a state is used for timing of processes. These three notions are covered by elements in the Network-Oriented Modelling approach based on temporal-causal networks, and form the defining part of a conceptual representation of a specific temporal-causal network model:

- **Strength of a connection $\boldsymbol{\omega}_{X,Y}$**
  Each connection from a state X to a state Y has a *connection weight value* $\boldsymbol{\omega}_{X,Y}$ representing the strength of the connection, often between 0 and 1, but sometimes also below 0 (negative effect) or above 1.
- **Combining multiple impacts on a state $\mathbf{c}_Y$(..)**
  For each state (a reference to) a *combination function* $\mathbf{c}_Y$(..) is chosen to combine the causal impacts of other states on state Y.
- **Speed of change of a state $\boldsymbol{\eta}_Y$**
  For each state Y a *speed factor* $\boldsymbol{\eta}_Y$ is used to represent how fast a state is changing upon causal impact.

Combination functions can have different forms, as there are many different approaches possible to address the issue of combining multiple impacts. The applicability of a specific combination rule for this may depend much on the type of application addressed, and even on the type of states within an application. Therefore, the Network-Oriented Modelling approach based on temporal-causal networks incorporates for each state, as a kind of label or parameter, a way to specify how multiple causal impacts on this state are aggregated. For this aggregation a number of standard combination functions are made available as options and a number of desirable properties of such combination functions have been identified (see [15], Chap. 2, Sects. 2.6 and 2.7), some of which are the scaled sum function and the advanced logistic sum function:

$$\mathbf{ssum}_\lambda(V_1, \ldots, V_k) = (V_1 + \ldots + V_k)/\lambda \quad \text{with } \lambda > 0$$

$$\mathbf{alogistic}_{\sigma,\tau}(V_1, \ldots, V_k) = [(1/(1 + \mathbf{e}^{-\sigma(V_1 + .. + V_k - \tau)}) - (1/(1 + \mathbf{e}^{\sigma\tau}))](1 + \mathbf{e}^{-\sigma\tau}) \quad \text{with } \sigma,\ \tau \geq 0$$

The above three concepts (connection weight, speed factor, combination function) can be considered as parameters or labels representing characteristics in a network model. In Fig. 1 a graphical conceptual representation of the temporal-causal network model presented here is shown. Adding the three above-mentioned types of labels to such a picture as shown in Fig. 1 provides a labelled graph as the model specification. In a non-adaptive network model these parameters are fixed over time. But to model processes by adaptive networks, not only the state levels, but also the values of some of these parameters can change over time.

The explanation of the depicted states can be found in Table 1. The two persons involved are indicated by the dotted boxes: A is the considered female Arnie with HSP, and B a normal person Bert. The states not in the boxes are world states, $\mathrm{ws}_{avoiding,s}$ one for Arnie's avoiding gaze, ws_s one for the stimulus s and ws_b one for the body state b. The gaze state has effect on what A represents, and (through observation) it is represented by B. The body state b of A is represented by both. In this way the two persons interact via the world states, in addition to communication between them, which is modeled by the connections from $esc_{B,b}$ to $rsc_{A,B}$, and from $esc_{A,n}$ to $rsc_{B,A}$.

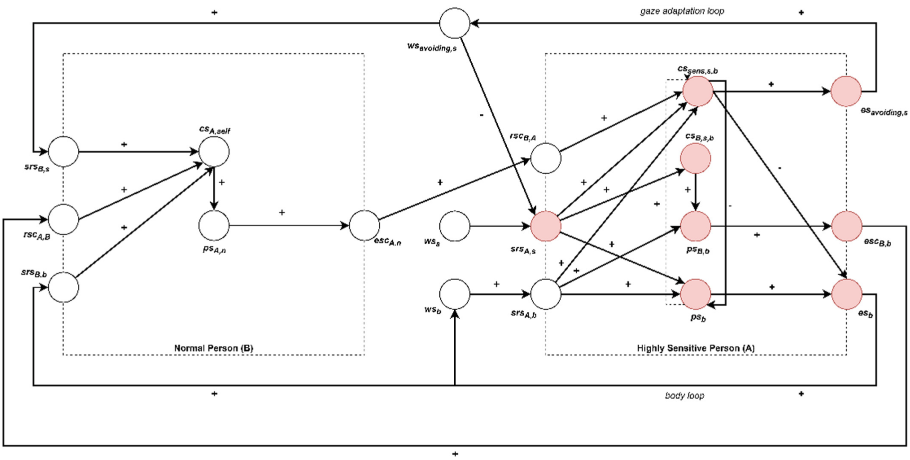


Fig. 1. Graphical conceptual representation of the network-oriented agent model.

Table 1. Overview of the states used in the model and their explanation

State	Explanation
$srs_{A,s}$	Sensory representation state of stimulus s by A
$srs_{B,s}$	Sensory representation state of avoiding gaze of A by B
$srs_{X,b}$	Sensory representation state of body state b by X (= A or B)
$rsc_{B,A}$	Representation state of A for communication from B
$rsc_{A,B}$	Representation state of B for communication from A
ps_b	Preparation state for body state b by A
$ps_{B,b}$	Preparation state for communication b of person A to person B
$ps_{A,n}$	Preparation state for communication n of person B to person A
$cs_{B,s,b}$	Control state for preparation state for communication by A of b to B
$cs_{A,self}$	Control state for self-other distinction of B concerning A
$cs_{sens,s}$	Control state for enhanced sensory sensitivity for s
es_b	Execution state for body state b
$esc_{B,b}$	Execution state for communication of body state b to B
$esc_{A,n}$	Execution state for communication of n to A
$es_{avoiding,s}$	Execution state for avoidance of stimulus s
ws_s	World state for stimulus s
ws_b	World state for body state b
$ws_{avoiding,s}$	World state for gaze avoiding stimulus s

The impact from one state on another state can have different values, positive or negative, usually in the interval [−1, 1]. These are some of the impacts:

- The presence ws_s of the stimulus has a positive influence on the sensory representation state presentation $\mathrm{srs}_{A,s}$ of the stimulus by Arnie.
- When Arnie has representation $\mathrm{srs}_{A,s}$ active, the control state $\mathrm{cs}_{sens.s,b}$ of the body state of Arnie for enhanced sensory sensitivity of the stimulus gets a positive impact, as well as the control state $\mathrm{cs}_{B,s,b}$ for body state for self-other distinction concerning Bert and the preparation state ps_b for the body state of Arnie get a positive impact.
- The body control state $\mathrm{cs}_{sens,s,b}$ of Arnie (attempting to regulate the enhanced sensory sensitivity), will stimulate the action $\mathrm{es}_{avoiding,s}$ of avoiding the stimulus with her gaze and will have a positive influence of the expression es_b of Arnie's body state.
- The body control state $\mathrm{cs}_{sens,s,b}$ of Arnie also has a negative influence on the body state expression es_b of Arnie, and also on the preparation ps_b; because Arnie is a sensitive person, Arnie has difficulties to suppress her body state expression, so this negative impact may be too weak.
- The control state $\mathrm{cs}_{B,s,b}$ for self-other distinction will stimulate the preparation $\mathrm{ps}_{B,b}$ for communication of the body state of Arnie to Bert.
- The preparation ps_b of the body state of Arnie will have a positive effect on the expression es_b of the body state of Arnie.
- The avoidance $\mathrm{es}_{avoiding,s}$ of Arnie of his gaze to the stimulus, will have a positive effect on the world state $\mathrm{ws}_{avoiding,s}$ of avoiding the stimulus with her gaze.
- The world state $\mathrm{ws}_{avoiding,s}$ where Arnie avoids the siren with her gaze will have a suppressing influence on the sensory representation srs_s that Arnie receives of the stimulus.
- The dotted arrow from ps_b to $\mathrm{cs}_{sens,s,b}$ indicates the connection on which Hebbian learning is applied. This connection may not be very strong initially, which makes the control state $\mathrm{cs}_{sens,s,b}$ not very active in case of high levels of ps_b (emotional response on the stimulus). By learning, the connection can become stronger which enables more effective control (by suppressing them stronger) of the emotional responses ps_b and es_b on the stimulus.

A conceptual representation of a temporal-causal network model can be transformed in a systematic or even automated standard manner into an equivalent numerical representation of the model as follows [14], Chap. 2:

- at each time point t each state Y in the model has a real number value in the interval [0, 1], denoted by $Y(t)$
- at each time point t each state X connected to state Y has an impact on Y defined as $\mathbf{impact}_{X,Y}(t) = \boldsymbol{\omega}_{X,Y}\, X(t)$ where $\boldsymbol{\omega}_{X,Y}$ is the weight of the connection from X to Y
- The *aggregated impact* of multiple states X_i on Y at t is determined using a *combination function* $\mathbf{c}_Y(..)$:

$$\begin{aligned}\mathbf{aggimpact}_Y(t) &= \mathbf{c}_Y(\mathbf{impact}_{X_1,Y}(t),\ldots,\mathbf{impact}_{X_K,Y}(t)) \\ &= \boldsymbol{c}_Y(\boldsymbol{\omega}_{X_1,Y}\boldsymbol{X_1}(\boldsymbol{t}),\ldots,\boldsymbol{\omega}_{X_K,Y}\boldsymbol{X_k}(\boldsymbol{t}))\end{aligned}$$

where X_i are the states with connections to state Y

- The effect of $\mathbf{aggimpact}_Y(t)$ on Y is exerted over time gradually, depending on speed factor η_Y:

$$Y(t+\Delta t) = Y(t) + \boldsymbol{\eta}_Y[\mathbf{aggimpact}_Y(t) - Y(t)]\Delta t$$

or

$$\mathbf{d}Y(t)/\mathbf{d}t = \boldsymbol{\eta}_Y[\mathbf{aggimpactY}(\mathbf{t}) - Y(t)]$$

- Thus, the following *difference* and *differential equation* for Y are obtained:

$$Y(t+\Delta t) = Y(t) + \boldsymbol{\eta}_Y[\mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}X_1(t),\ldots,\boldsymbol{\omega}_{X_k,Y}X_k(t)) - Y(t)]\,\Delta t$$

$$\mathbf{d}Y(t)/\mathbf{d}t = \boldsymbol{\eta}_Y[\mathbf{c}_Y(\boldsymbol{\omega}_{X_1,Y}X_1(t),\ldots,\boldsymbol{\omega}_{X_k,Y}X_k(t)) - Y(t)]$$

As explained above in a general setting, from the conceptual representation of a temporal-causal network model (including labels for connection weights $\boldsymbol{\omega}_{X,Y}$, combination functions $\mathbf{c}_Y(\ldots)$ and speed factors $\boldsymbol{\eta}_Y$), the difference and differential equations of the numerical representation can be easily generated. For the key states the difference equations are:

- For any point in time t for state $\mathrm{srs}_{A,s}$

$$\mathrm{srs}_{A,s}(t+\Delta t) = \mathrm{srs}_{A,s}(t) + \eta_{\mathrm{srs}_s}[\mathrm{c}_{\mathrm{srs}_s}(\omega_{\mathrm{ws}_s,\mathrm{srs}_s}\mathrm{ws}_s(t)) - \mathrm{srs}_{A,s}(t)]\Delta t$$

- For any point in time t for state $\mathrm{cs}_{sens,s}$

$$\begin{aligned}\mathrm{cs}_{sens,s,b}(t+\Delta t) = \mathrm{cs}_{sens,s,b}(t) + \eta_{\mathrm{cs}_{sens,s,b}}\,[\mathrm{c}_{\mathrm{cs}_{sens,s,b}}(\omega_{\mathrm{rsc}_{B,A},\mathrm{cs}_{sens,s,b}}\mathrm{rsc}_{sens,s}(t),\\ \omega_{\mathrm{srs}_{A,S},\mathrm{cs}_{sens,s,b}}\mathrm{srs}_{A,s}(t), \omega_{\mathrm{srs}_{A,b},\mathrm{cs}_{sens,s,b}}\mathrm{srs}_{A,b}(t), \omega_{\mathrm{ps}_b,\mathrm{cs}_{sens,s,b}}\mathrm{ps}_b(t)) - \mathrm{cs}_{sens,s,b}(t)]\,\Delta t\end{aligned}$$

- For any point in time t for state $\mathrm{cs}_{B,s,b}$

$$\mathrm{cs}_{B,s,b}(t+\Delta t) = \mathrm{cs}_{B,s,b}(t) + \eta_{\mathrm{cs}_{B,s,b}}[\mathrm{c}_{\mathrm{srs}_s}(\omega_{\mathrm{srs}_{A,s},\mathrm{cs}_{B,s,b}}\mathrm{srs}_{A,s}(t)) - \mathrm{cs}_{B,s,b}(t)]\Delta t$$

- For any point in time t for state $\mathrm{ps}_{B,b}$

$$\begin{aligned}\mathrm{ps}_{B,b}(t+\Delta t) = \mathrm{ps}_{B,b}(t) + \eta_{\mathrm{ps}_{B,b}}[\mathrm{c}_{\mathrm{srs}_s}(\omega_{\mathrm{srs}_{A,b},\mathrm{ps}_{B,b}}\mathrm{srs}_{A,b}(t), \omega_{\mathrm{cs}_{B,s,b},\mathrm{ps}_{B,b}}\mathrm{cs}_{B,s,b}(t))\\ - \mathrm{ps}_{B,b}(t)]\Delta t\end{aligned}$$

- For any point in time t for state ps_b

$$\mathrm{ps}_b(t+\Delta t) = \mathrm{ps}_b(t) + \eta_{\mathrm{ps}_b}[\mathrm{c}_{\mathrm{srs}s}(\omega_{\mathrm{srs}_{A,b},\mathrm{ps}_b}\mathrm{srs}_{A,b}(t), \omega_{\mathrm{srs}_{A,s},\mathrm{ps}_b}\mathrm{srs}_{A,s}(t), \omega_{\mathrm{cs}_{sens,s},\mathrm{ps}_b}\mathrm{cs}_{sens,s}(t)) - \mathrm{ps}_b(t)]\Delta t$$

- For any point in time t for state $\mathrm{es}_{avoiding,s}$

$$\mathrm{es}_{avoiding,s}(t+\Delta t) = \mathrm{es}_{avoiding,s}(t) + \eta_{\mathrm{es}_{avoiding,s}}[\mathrm{c}_{\mathrm{es}_{avoiding,s}}(\omega_{\mathrm{cs}_{sens,s,b},\mathrm{es}_{avoiding,s}}\mathrm{cs}_{sens,s,b}(t)) - \mathrm{es}_{avoiding,s}(t)]\Delta t$$

- For any point in time t for state $\mathrm{esc}_{B,b}$

$$\mathrm{esc}_{B,b}(t+\Delta t) = \mathrm{esc}_{B,b}(t) + \eta_{\mathrm{esc}_{B,b}}[\mathrm{c}_{\mathrm{esc}_{B,b}}(\omega_{\mathrm{ps}_{B,b},\mathrm{esc}_{B,b}}\mathrm{ps}_{B,b}(t)) - \mathrm{esc}_{B,b}(t)]\,\Delta t$$

- For any point in time t for state es_b

$$\mathrm{es}_b(t+\Delta t) = \mathrm{es}_b(t) + \eta_{\mathrm{es}_b}[\mathrm{c}_{\mathrm{esc}_{B,b}}(\omega_{\mathrm{cs}_{sens,s,b},\mathrm{es}_b}\mathrm{cs}_{sens,s,b}(t), \omega_{\mathrm{ps}_b,\mathrm{es}_b}\mathrm{ps}_b(t)) - \mathrm{es}_b(t)]\Delta t$$

The weight $\omega = \omega_{\mathrm{ps}b,\mathrm{cs}sens,s}$ of the connection from ps_b to $\mathrm{cs}_{sens,s}$ to which Hebbian learning is applied is dynamic and can be handled as if it is a state; the following difference equation is used:

$$\omega(t+\Delta t) = \omega(t) + \eta_\omega[\mathrm{c}_\omega(\mathrm{ps}_b(t), \mathrm{cs}_{sens,s,b}(t), \omega(t)) - \omega(t)]\,\Delta t$$

with speed factor (learning rate) η_ω, and combination function

$$\mathrm{c}_\omega(V_1, V_2, W) = V_1 V_2(1-W) + \mu\, W$$

Here μ is a persistence parameter with values between 0 and 1, where $\mu = 1$ means fully persistent, and $\mu < 1$ indicates some extent of extinction. So,

$$\mathrm{c}_\omega(\mathrm{ps}_b(t), \mathrm{cs}_{sens,s,b}(t), \omega(t)) = \mathrm{ps}_b(t)\,\mathrm{cs}_{sens,s,b}(t)\,(1-\omega(t)) + \mu\,\omega(t)$$

and

$$\omega(t+\Delta t) = \omega(t) + \eta_\omega[\mathrm{ps}_b(t)\mathrm{cs}_{sens,s,b}(t)\,(1-\omega(t)) - (1-\mu)\,\omega(t)]\,\Delta t$$

4 Simulation Results

An estimation was made about which choices for the parameters (connection weights, combination functions, speed factors) to use, and initial values for the states were chosen. The connection weights can be found in the Table 2. All speed factors were 0.25. For all states with a single incoming connection the identity combination function has been used: **id**(V) = V. For states $\mathrm{ps}_{B,b}$ and $\mathrm{cs}_{A,self}$ the scaled sum combination function has been used with scaling factor 2 and 3, respectively. For states $\mathrm{srs}_{A,s}$, ps_b,

$cs_{sens,s,b}$ and es_b the advanced logistic combination function has been used with steepness and threshold combinations (σ, τ) = (4.137, 0.139), (2.402, 0.3), (2, 1.5), (5, 0.4), respectively. The stimulus $ws_{,s}$ was constant 1, the other states initially have value 0. The learnt connection initially is 0.1, and Δt was 1. In Fig. 2 (left hand side) a graph of the resulting state values can be seen, whereas the right hand side shows a graph of the learning. There are 19 lines, which stand for the 19 different states. However, in the graph, only 18 lines are visible. This is due to the fact that line 4 and line 16 share the same values and therefore line 16 overlaps line 4, which makes line 4 not visible in the graph. According to the scenario, the lines of the states in this graph are not fully satisfactory yet as they do not show the right sequence of events and they also do not represent the fluctuation of the states as expected. Therefore, further parameter tuning was needed to get the lines of the states as expected according to the agent model with the assigned plus and minus weights.

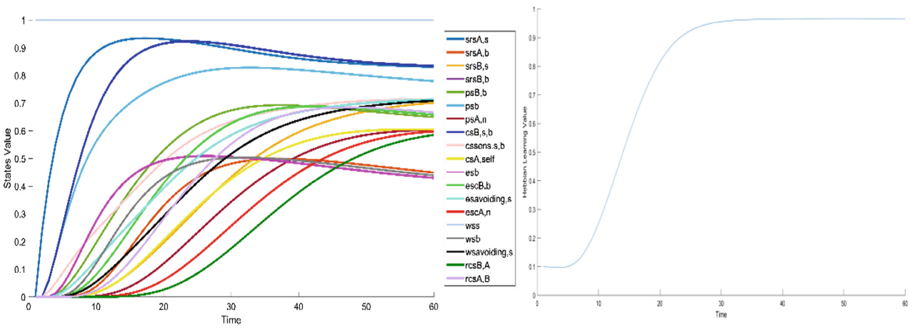


Fig. 2. A first example simulation of the agent model: states (left graph) and the adaptive connection (right graph) based on Hebbian learning

Table 2. Connection weights in the example simulations

from\to	$srs_{A,s}$	$srs_{A,b}$	$srs_{B,s}$	$srs_{B,b}$	$rsc_{B,A}$	$rsc_{A,B}$	ps_b	$ps_{B,b}$	$ps_{A,n}$	$cs_{B,s,b}$	$cs_{A,self}$	$cs_{sens,s,b}$	es_b	$esc_{B,b}$	$esc_{A,n}$	$es_{avoiding,s}$	ws_s	ws_b	$ws_{avoiding,s}$
$srs_{A,s}$							1			1		0.3							
$srs_{A,b}$								1				1							
$srs_{B,s}$											1								
$srs_{B,b}$											1								
$rsc_{B,A}$												1							
$rsc_{A,B}$											1								
ps_b												0.1	1						
$ps_{B,b}$														1					
$ps_{A,n}$															1				
$cs_{B,s,b}$								1											
$cs_{A,self}$									1										
$cs_{sens,s,b}$							-0.9						-0.5			1			
es_b				1														1	
$esc_{B,b}$						1													
$esc_{A,n}$					1														
$es_{avoiding,s}$																			1
ws_s	1																1		
ws_b		1																	
$ws_{avoiding,s}$	-0.5		1																

After more advanced tuning of the parameters (an explanation of how that was done can be found in Sect. 5), a second example simulation was obtained; see Fig. 3.

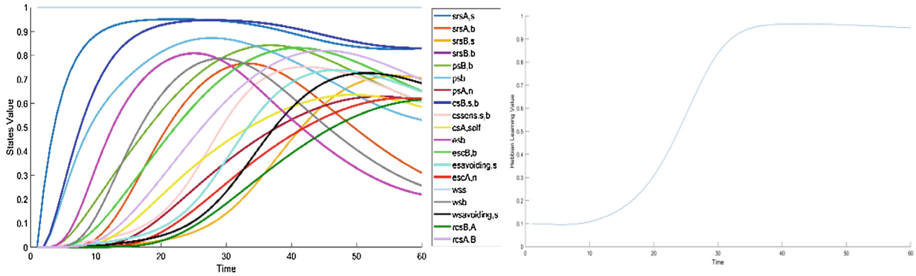


Fig. 3. Second example simulation of the agent model: states (left graph) and the adaptive connection (right graph) based on Hebbian learning

Both simulation examples show the Hebbian learning process of person A. The Hebbian learning connection is already present from the start, however initially it is low: 0.1. After person A has an interaction with Person B, Person A will learn by making the Hebbian learning connection stronger, until it is around 1. The next time person A faces an upsetting external stimulus, person A will be able to handle the external stimulus all by him or herself. In the second example simulation in Fig. 4 it can be seen that the line slowly lowers after some time, this can be explained by the fact that learning strength lowers after some time due to lower state values for ps_b and $cs_{sens,s,b}$, and therefore there is some extinction.

5 The Role of Requirements and Parameter Tuning

In this section it is described how a requirement for the expected pattern for the simulation was identified and used in automated parameter tuning based on Simulated Annealing. As mentioned in Sect. 4, the order of activations in the first example simulation was not satisfactory. Therefore, based on [15] a requirement was identified for the order of activation. This requirement was expressed as a form of use case in Table 3. Note that the notions of requirement and use case were borrowed from the Software Engineering area, and turned out useful in this case, in particular in combination with parameter tuning.

Table 3. Required succession of activation of the states: use case for the model

Time	1	2	3	4	5	6	7	8	9	10
State	ws_s	$srs_{A,s}$	$cs_{sens,s,b}$ $cs_{B,s,b}$ ps_b	es_b $ps_{B,b}$ $es_{avoiding,s}$	$esc_{B,b}$ $ws_{avoiding,s}$ ws_b $srs_{B,b}$	$rsc_{A,B}$ $srs_{A,b}$ $cs_{A,self}$ $srs_{B,s}$	$ps_{A,n}$	$esc_{A,n}$	$rsc_{B,A}$	$cs_{sens,s,b}$

This use case was expressed in activation numbers for the states that show how after some point in time from the value 0 the activation level of a state becomes high and later lower. These numbers were used as a form of pseudo-empirical data in the parameter tuning process for the speed factor parameters based on Simulated Annealing. During the tuning process the sum of squared residues SSR was used as an error function. The pattern over the number of iterations is shown in Fig. 4. The parameter values found are shown in Table 4. The graphs for these values are the ones shown in Fig. 3 above. The minimal error SSR found for these speed factors is 117.2719. The number of data points used is $N = m * n$; where m is the number of states (19) and n the number of time points used per state (60), so N = 19 * 60 = 1140, and SSR/N = 0.07. The square root of this is an indication for the average deviation, this is 0.26. This value could be considered a bit high; however, recall that the requirement was the focus, and the requirement has been satisfied by the tuning process. There was no further requirement that number wise the empirical data should be approximated. Therefore, by satisfying the requirement, the tuning process served our purposes.

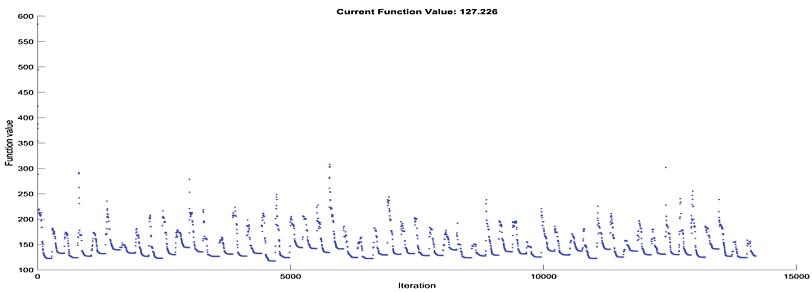


Fig. 4. Graph of the sum of squares SSR error over the number of iterations

Table 4. Speed factor values after parameter tuning

State	$srs_{A,s}$	$srs_{A,b}$	$srs_{B,s}$	$srs_{B,b}$	$rsc_{B,A}$	$rsc_{A,B}$	ps_b	$ps_{B,b}$	$ps_{A,n}$	$cs_{B,s,b}$	$cs_{A,self}$	$cs_{sens,s,b}$
Speed factor	0.359	0.5	0.498	0.368	0.412	0.474	0.5	0.469	0.393	0.493	0.493	0.498
State	es_b	$esc_{B,b}$	$esc_{A,n}$	$es_{avoiding,s}$	ws_s	ws_b	$ws_{avoiding,s}$					
Speed factor	0.5	0.473	0.473	0.452	0.498	0.5	0.487					

6 Verification of the Agent Model by Mathematical Analysis

In this section it is shown how Mathematical Analysis was used to verify whether the implemented model is in accordance with the model specification. To this end stationary points of the following states were identified and analysed: es_b, ps_b, $cs_{sens,s,b}$, $cs_{B,s,b}$ and $esc_{B,b}$. When Y is a state, Y has a stationary point at t if $\mathbf{d}Y(t)/\mathbf{d}t = 0$. For a temporal-causal network there is a simple and very effective criterion for having a stationary point: state Y has a stationary point at $t \Leftrightarrow$ $\mathbf{aggimpact}_Y(t) = Y(t) \Leftrightarrow \mathbf{c}_Y(\boldsymbol{\omega}_{X1,Y}X_1(t), \ldots, \boldsymbol{\omega}_{Xk,Y}X_k(t)) = Y(t)$ where X_i are the states with connections to state Y. To apply this, from a simulation for each of the five states state a time point was identified at which a stationary point occurred (in particular, a maximum): see Table 4, first and second row; the state values at these time points are in the third row. Next, for each of the columns, for these time points the relevant aggregated impact was calculated based on the combination function, the connection weights and the state values used in this aggregated impact: see the fourth row in Table 5. Finally, in the fifth row the absolute deviation between aggregated impact and state value is shown: $|\mathbf{aggimpact}_Y(t) - Y(t)|$.

Table 5. Verification of the model by analyzing stationary points

Maxima	es_b	ps_b	$cs_{sens,s,b}$	$cs_{B,s,b}$	$esc_{B,b}$
Time-point	25	28	42	28	42
State value	0.808715	0.872455	0.752271	0.947237	0.831066
Aggregated impact	0.803666	0.867326	0.745274	0.946509	0.826951
Absolute deviation <0.01	0.005049	0.005129	0.006997	0.000728	0.004115

As can be seen above, all five states have an absolute deviation below 0.01, which provides evidence that the implemented model does what is expected.

7 Discussion

The human-like agent model introduced in this paper was designed as a temporal-causal network based on the Network-Oriented Modeling approach described in [14]. This approach allows to model adaptive and cyclic processes based on causal relations. The network-oriented agent model designed incorporates mechanisms found in literature from Neuroscience such as [11, 12]. Such mechanisms determined the basic architecture of the network. The network model was able to reproduce the pattern what is expected from [11].

Such a model has a relatively high number of parameters. Values for such parameters cannot be obtained from the literature; they have to be estimated in one way or the other. Doing that ‘by hand’ can be a quite elaborate process. Therefore, a different approach was chosen through which there is automated support for this process. Using notions known from Software Engineering, the expected pattern was formulated as a requirement, for which a (qualitative) use case was generated. For this use case, numbers were created to make numerical methods applicable. The method used was Simulated Annealing. Indeed, the solution found satisfies the requirement: it generates the expected pattern. So, in this case this approach via requirements and use cases in combination with automated parameter tuning was quite helpful.

Sometimes it is suggested that by parameter tuning you can generate any pattern from any model if suitable parameter values are used. In general, this is not true, as long as a specific architecture is used, as in this case, based on literature [11], in order to get a human-like model. Then what is shown is that in particular this architecture, which is justified by [11], is able to simulate the expected pattern. That this architecture can also generate other patterns and that also other architectures can generate the same pattern might be true but is not relevant.

The introduced computational model can be applied as a basis for a virtual training environment for therapists to get more insight in the processes taking place in highly sensitive persons. As another step to enhance the value of such an environment, the model may be extended by a number of possible therapies and their effects on these processes.

Acknowledgement. We would like to express our gratitude to Fawad Taj for his support for the Matlab Network-Oriented Modeling template in which the model was implemented.

References

1. Jagiellowicz, J., Aron, A., Aron, E.N.: Relationship between the temperament trait of sensory processing sensitivity and emotional reactivity. Soc. Behav. Pers. Int. J. **44**(2), 185–199 (2016)
2. Aron, E.N.: High sensitivity as one source of fearfulness and shyness: preliminary research and clinical implications. In: Schmidt, L.A., Schulkin, J. (eds.) Extreme Fear, Shyness, and Social Phobia: Origins, Biological Mechanisms, and Clinical Outcomes, pp. 251–272. Oxford University Press, New York (1999)
3. Belsky, J., Pluess, M.: Beyond diathesis stress: differential susceptibility to environmental influences. Psychol. Bull. **135**(6), 885–905 (2009)
4. Aron, E.N.: Counseling the highly sensitive person. Couns. Hum. Dev. **28**(9), 1–7 (1996)
5. Aron, E.N.: The Highly Sensitive Child: Helping Our Children Thrive When the World Overwhelms Them. Harmony, New York (2002)
6. Aron, E.N.: The Highly Sensitive Person in Love. Harmony, New York (2000)
7. Rytovaara, M., Aron, E.N.: The Highly Sensitive Person. How to Thrive When the World Overwhelms You. Thorsons, London (1999)
8. Aron, E.N., Aron, A., Jagiellowicz, J.: Sensory processing sensitivity: a review in the light of the evolution of biological responsivity. Pers. Soc. Psychol. Rev. **16**(3), 262–282 (2012)
9. Aron, E.N., Aron, A.: Sensory-processing sensitivity and its relation to introversion and emotionality. J. Pers. Soc. Psychol. **73**(2), 345–368 (1997)
10. Benham, G.: The highly sensitive person: stress and physical symptom reports. Pers. Individ. Differ. **40**(7), 1433–1440 (2006)
11. Hebb, D.O.: The Organization of Behavior: A Neuropsychological Theory. Psychology Press, London (2005)
12. Munakata, Y., Pfaffly, J.: Hebbian learning and development. Dev. Sci. **7**(2), 141–148 (2004)
13. Jagiellowicz, J., Xu, X., Aron, A., Aron, E.N., Cao, G., Feng, T., et al.: The trait of sensory processing sensitivity and neural responses to changes in visual scenes. Soc. Cogn. Affect. Neurosci. **6**(1), 38–47 (2011)
14. Kim, J.: Philosophy of Mind. Westview Press, Boulder (1996)
15. Treur, J.: Network-Oriented Modeling: Addressing Complexity of Cognitive, Affective and Social Interactions. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45213-5

16. Kuipers, B.J.: Commonsense reasoning about causality: deriving behavior from structure. Artif. Intell. **24**, 169–203 (1984)
17. Kuipers, B.J., Kassirer, J.P.: How to discover a knowledge representation for causal reasoning by studying an expert physician. In: Proceedings of the 8th International Joint Conference on Artificial Intelligence, IJCAI 1983, pp. 49–56. William Kaufman, Los Altos (1983)
18. Pearl, J.: Causality. Cambridge University Press, Cambridge (2000)

Check for updates

SAIL: A Social Artificial Intelligence Layer for Human-Machine Teaming

Bob van der Vecht(✉), Jurriaan van Diggelen, Marieke Peeters, Jonathan Barnhoorn, and Jasper van der Waa

TNO, The Hague, The Netherlands
{bob.vandervecht, jurriaan.vandiggelen, marieke.peeters, jonathan.barnhoorn, jasper.vanderwaa}@tno.nl

Abstract. Human-machine teaming (HMT) is a promising paradigm to approach future situations in which humans and autonomous systems closely collaborate. This paper introduces SAIL, a design method and framework for the development of HMT-concepts. Starting point of SAIL is that an HMT can be developed in an iterative process in which an existing autonomous system is enhanced with social functions tailored to the specific context. The SAIL framework consists of a modular social layer between autonomous systems and human team members, in which all social capabilities can be implemented to enable teamwork. Within SAIL, HMT-modules are developed that construct these social capabilities. The modules are reusable in multiple domains.

Next to introducing SAIL we demonstrate the method and framework using a proof of concept task, from which we conclude that the method is a promising approach to design, implement and evaluate HMT-concepts.

Keywords: Human-machine teaming · Social artificial intelligence
Collaboration · Design method

1 Introduction

It is hard to imagine a future society without cognitive robots and agents. Artificial systems will fulfil all kinds of societal roles, e.g. personal assistant, tutor, companion, and coach [1]. Rest aside these futuristic developments, autonomous vehicles in civilian transport and autonomous robots for the military are already here. As the integration of autonomous systems in everyday life is increasing the complexity of human-machine interaction, meaningful interaction and collaboration between humans and agents becomes a pressing problem. This has led to an increased interest in this topic, and initiated debates discussing the balance between human control and system autonomy. For example, the discussion on *meaningful human control* [2] addresses a growing concern about system autonomy potentially leading to conflicts with humanitarian law and ethics. The question arises how autonomous systems can support interaction with humans in a way that ultimately allows humans to stay in control.

Human-machine teaming (HMT) is a commonly proposed paradigm to realize this [3, 4]. The challenge of implementing an agent as a team member is often mistakenly

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 262–274, 2018.
https://doi.org/10.1007/978-3-319-94580-4_21

construed as simply *placing a user interface on top of an autonomous system*. In fact, it requires a whole layer of artificial intelligence that encompasses a range of social capabilities enabling the autonomous system to behave in a observable, predictable, and directable way [5].

Whereas current autonomous systems are typically designed to meet the objectives of a specific problem in a specific domain, collaborative team members must be designed also to be part of human-machine teams and require a number of generic social capabilities, for example:

- explaining their behavior in a way that is understandable to humans [6]
- proactively sharing information within the team based on the information needs [8]
- growing towards more mature teams, i.e. human-agent team development [4]

The requirements regarding social capabilities and team functions strongly interact with the properties of both human and artificial team members, as well as with the task environment. For example, if the human cannot do some tasks well in specific circumstances, they must be transferred to the machine. Furthermore, the social capabilities depend on the capabilities of the autonomous system. For example, if the system is able to autonomously adapt its behavior in reaction to the workload of an operator, then the system should be provided with the workload information, whereas a less sophisticated system does not need this. Also the task environment may shape the type of collaboration, e.g. spatial constraints may limit the amount of interactions. Each of these different perspectives must be integrated when designing the HMT. Figure 1 shows how the multiple disciplines come together in defining and implementing collaboration in HMT.

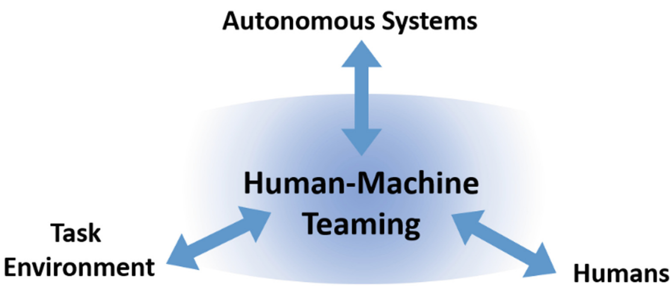


Fig. 1. The HMT requirements for a specific problem are influenced by factors from multiple disciplines: human factors, autonomous systems and the task environment.

Most research on human-machine teaming focuses on the one or two of the relevant topics. Human factors research, for instance, focuses on the relevant properties of HMT from a user perspective, such as trust in automation [7], and shared situation awareness [8]. Based on these results, a few requirements engineering methodologies for HMT have been proposed (e.g. [5]), but they do not provide a system architecture. To implement HMT systems based on these requirements, the engineer could employ techniques from the multi-agent systems (MAS) community (such as policy-based systems [9], agent organizations [10]). However, these agent-based frameworks have been developed with a focus on multi-agent interaction, and do not address particular human factors issues present in HMT. Furthermore, testing methods for MAS (such as [11]), do not provide ways to perform human-in-the-loop tests.

Whereas HMT contains generic properties, a one-size fits all solution for HMT does not exist, as the types of tasks, humans and autonomous systems is different per context. This motivates the need for a method to develop, implement and evaluate HMT in a modular way. As solution we introduce the SAIL (Social Artificial Intelligence Layer) method and framework. The purpose of SAIL is to integrate the human factors and autonomous systems in one design method that is supported by a software framework for implementing and testing HMT-concepts. Starting point of SAIL is that an HMT is developed as an additional layer of social intelligence which complements the autonomous capabilities of the system. The social capabilities to interact and collaborate with the autonomous systems are made available by connecting the system to the SAIL framework.

The SAIL method consists of an iterative development cycle of four steps (Fig. 2). The approach is based on solid systems engineering practices, such as rapid prototyping, short implementation-test cycles, component reuse, and user-centered design.

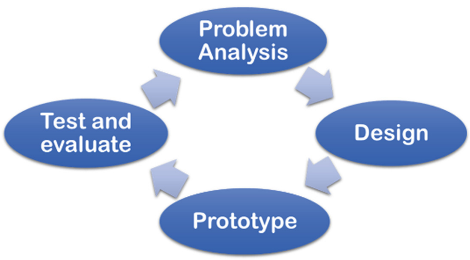


Fig. 2. The SAIL method to develop HMT consists of four phases

With SAIL we facilitate the design and development of reusable components for the most common HMT requirements in the following aspects that are considered relevant:

- It enables in-house development of artificial social intelligence that can be used on top of closed-source systems.
- They can be applied to a variety of autonomous systems with different underlying architectures (i.e. independent of the underlying control logic [12]).
- It allows for evaluating team performance, to iteratively improve HMT-concepts and to facilitate longitudinal team development, as team behavior changes over time [13].

SAIL components interact with each other via human-readable standardized communication languages and protocols. Using SAIL, social requirements for autonomous systems to properly function in HMT can be defined. In the end these requirements can be adopted in the development of autonomous systems themselves by their producers. The paper will further describe the SAIL framework and the development method.

2 SAIL: A Social Artificial Intelligence Layer

2.1 Modular HMT Approach

The base idea behind SAIL is that the autonomous systems that need to be incorporated in the HMT already exist, and we minimize modifications to their internal behavior. Our HMT development approach aims at establishing a social layer between humans and autonomous systems that allows them to function as a coherent team. For this layer we use a modular approach where a single HMT-module facilitates a specific desired aspect for the HMT. Examples are a module to provide status information, or a module to enforce behavior policies. Each module is a single software component that receives certain input and provides a specific output. Modules can be linked in parallel and sequence, as such they can provide each other with enriched inputs.

Figure 3 shows an example of a modular SAIL. On the left are the human users and on the right the autonomous systems. The HMT-modules are situated in between, receiving their input either from the systems or the users.

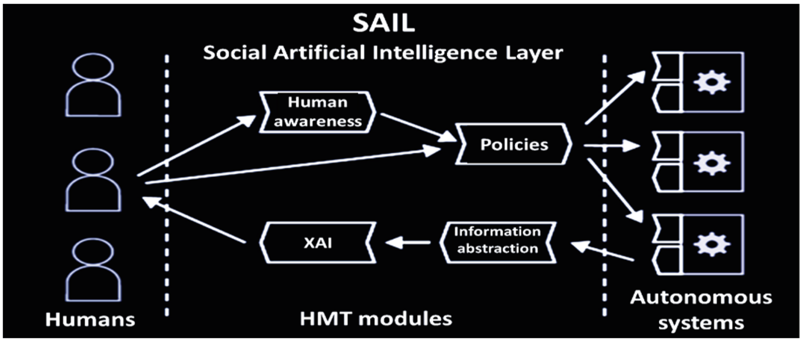


Fig. 3. The modular approach to a social AI layer

This modular approach requires three key components; (1) a library of HMT-modules, (2) a language in which modules and systems can communicate and (3) the ability of autonomous systems to provide internal information and receive external control. In the next paragraphs we elaborate these components of SAIL.

2.2 Communication Language

The actors and modules in the SAIL framework need to communicate and understand each other. As part of SAIL, we introduce HATCL (Human-Agent-Teaming Communication Language), a communication language specifically designed for human machine teaming. HATCL therefore has a central place in the SAIL framework.

The concept of communication languages has been studied since the nineties [14]. For example, FIPA-ACL [15] was developed with the aim to standardize communication in multi-agent systems. Whereas the use of FIPA-ACL decreased in recent years, we partly reuse its ideas in HATCL. HATCL is a language which is used by the different agents (users, autonomous systems and HMT-modules) to communicate. HATCL specifies three main elements: the message format (including performatives), ontologies and protocols.

- Ontologies: following the ontology approach of [16], HATCL consists of a top-ontology laying out the HMT-concepts and message structure. Furthermore it includes domain ontologies for domain specific data structures.
- Performatives specify the intention of a message, such as *inform*, *request*, *confirm* and also *permissions* and *obligations* in a directive setting.
- Protocols are predefined conventions of message exchange to facilitate coordination between actors. E.g., the specification for the expected response to a request, or an interaction scheme to set up a contract for collaboration.

Key is that HATCL facilitates the communication between software, but also human operators. Therefore, the language has to be understandable for humans, and match with their mental concepts. Furthermore, it contains concepts that can be used to set up team functions for coordination and collaboration.

2.3 Interfacing with an Autonomous System

Another requirement for SAIL is the ability of the autonomous systems to interface with the SAIL framework via the HATCL ontology. We assume that the developers of the HMT do not have control about the internal software of the autonomous system. The autonomous systems may not have been designed for the social interaction, however, social capabilities become within reach when an interface with SAIL is created.

Within SAIL, all modules are able to communicate with one another using the HATCL, plus domain ontologies. Regarding the autonomous systems, “semantic anchors” are required that define how ontology concepts are linked to internal data structures and control logic of the autonomous system, as displayed in Fig. 4. The interface of autonomous systems and SAIL therefore consists of the specification of input ontologies (the information it accepts), output ontologies (the information it sends) and the semantic anchors. The semantic anchors form a crucial connection between the developers of the autonomous systems and the developers of the teaming functions. The semantic anchors are the only required adaptation to the autonomous system.

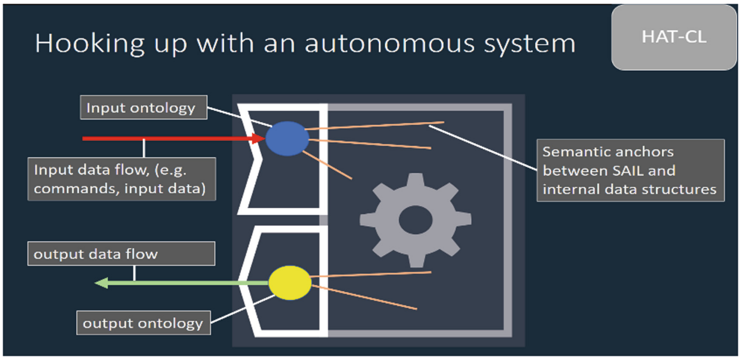


Fig. 4. Interfacing with an autonomous system requires input-output specifications using ontologies and semantic anchors that connect the inside of the autonomous system

3 Design Guidelines

This section provides guidelines for the four design phases of the SAIL method as shown in Fig. 2. The starting point of SAIL is that an HMT is constructed without changing the internals of the autonomous system. We are able, however, to define an interface to connect it to SAIL.

3.1 Problem Analysis

The key to the problem analysis is to describe the three influence factors shown in Fig. 1, i.e., the autonomous systems, the human factors and the task environment. This leads to constraints and requirements regarding social capabilities. The goal of this phase is to identify the gap between the social capabilities and needs of the autonomous system compared to the '*as is*'-situation, where no SAIL is added to the system. In the next step, the SAIL will be designed to overcome this capability gap. Dimensions that need to be explored for the three influence factors are:

- Autonomous systems: How many systems are involved? Do they differ in capabilities? What are the communication capabilities? Which ontologies are accepted?
- Humans: How many humans are involved? In what role? What is their skills (e.g., trained versus untrained)? Which information is required for their task?
- Task environment: What is the time scale, time pressure? Is it a routine task or does it require creativity? What are the risks of wrong decisions? What does the physical environment look like (e.g., physical distance, environment dynamics, uncertainty)?

Besides these three bullets, an important aspect to consider is the team phase of the HMT that will be designed and evaluated. Are they in a training phase where new HMT-concepts are tested, and the teaming functions are learned? Or is the team already trained and working with certain routines?

3.2 HMT Design

Based on the capability gap between collaborative needs of the team and the capabilities of the autonomous system, the SAIL will be designed. In a modular way components can be designed to meet these social requirements. The philosophy of our method is that these collaborative needs are generic to some extent, and therefore, it is possible to build a library of those HMT-modules. We have identified some categories of functions that support teamwork in HMT (this list does not aim to be a complete list):

- **Situation Awareness:** situation awareness functions share information within the team about the task progression and the environment. An example is a knowledge base that presents information about relevant environment context and position of team members. Such a module facilitates coordination of tasks within the team. To a human team member, a visual user interface might be connected to provide SA.
- **Human Awareness:** human awareness functions specifically disclose information about human team members. Technology is available to monitor presence, attention or trust of humans. If this information is made available to artificial team members, they can adjust their behavior or way of interacting.
- **Explainable AI:** explainable AI (XAI) functions specifically disclose information from artificial team members to human operators. This typically concerns more advanced information than the SA type of functions. It allows human team members to understand the rationale for the system's decisions. For example, an analyst who receives recommendations from an algorithm needs to understand why the algorithm has recommended certain activity.
- **Working agreements:** with working agreements we refer to general behaviour policies that should be followed by team members. In literature the other terms are also used: (social) norms [10], contracts or policies [9]. The working agreements define the boundaries within which team members may act autonomously. They may specify the applicable permissions or obligations, and interaction protocols.

The HMT-modules that are designed in this phase can be drawn in a network configuration to specify how they interact and exchange information among themselves and to the user enriched with details about the information exchange.

3.3 Prototype Implementation

SAIL comes with a software framework in which the autonomous system(s) and HMT-modules can implemented. Through the SAIL GUI, a human-agent team can be configured by defining the active actors, modules and the lines of communication between them. The framework, therefore, allows for rapid testing of different HMT configurations. A set of generic HMT-modules can be activated or deactivated, and for case-specific circumstances new modules can be implemented. Communication between modules is done via sockets. Therefore modules and systems that are implemented in different programming languages can all be connected.

Ontologies can be loaded into the SAIL framework. Typically a version of the HATCL is added, plus other specific domain ontologies if they are needed. The modules that are included in the configuration publish which ontologies they accepts. Therefore the validity of the HMT configuration can be checked.

3.4 Test and Evaluate

The SAIL framework is developed for human-in-the-loop testing and evaluation in order to facilitate an iterative development process and longitudinal team development, as teamwork may change over time when both human and artificial team members are getting used to one another. The fourth phase in the SAIL method is meant to set-up user experiments with the prototype implementation of HMT-concepts. Typically multi-criteria analyses are needed to assess team performance and usability aspects.

4 Proof of Concept

To illustrate the use of the SAIL method and framework with a proof of concept (PoC) task, we implemented a simulated futuristic defense task. In the task, the human operator takes on the role of a marine officer on board of a stationary naval vessel. His mission is to protect the ship from incoming hostile drones together with a swarm of autonomous unmanned aerial vehicles (UAVs) while minimizing collateral damage. In principle, the UAVs autonomously locate themselves around the ship and independently detect, identify (as friend or foe) and eliminate hostiles. The UAVs have a varying identification accuracy. The operator can improve team performance by supervising the task and prioritizing targets. Furthermore, the operator has access to external information regarding identification of contacts, which (s)he can interpret, whereas the UAVs cannot. The team task has the following properties that justify HMT:

- The UAVs are able to do the task autonomously, but they make mistakes.
- The human operator can control and supervise the UAV's, however, due to workload the operator cannot do everything by hand.
- The operator has knowledge that the autonomous system does not have.
- Human-in-the-loop solutions are typically preferred in this type of defense scenario.

We used this task environment to create a setting where a user repeatedly cycles through a mission simulation and after-action-review, allowing team development by optimizing the set of working agreements. The set of available HMT-modules determines how the user can affect the swarm's behavior.

The PoC served a number of goals. First, we gained experience with implementing a task using the SAIL framework. Second, we tried out the task environment, and whether it was sufficiently rich, realistic and challenging for evaluating HMT-concepts. Third, we tested our first, rudimentary implementation of the set of HMT-modules.

4.1 PoC Problem Analysis

The task can be characterized by: an unpredictable and dynamic environment, high differentiation within the team as the user and UAVs have very different skills; short repeated work cycles, allowing study of team development; high time criticality; and a high risk situation.

The UAV is able to autonomously execute the force protection task. Its behavior model is a state machine with three states: *surveillance*, *identification* and *engagement*. When operating in the swarm the UAV's do not communicate with each other. The UAVs have a varying identification accuracy, which leads to mis-identification, and to the elimination of neutral contacts, which is interpreted as collateral damage. The UAVs have no capabilities to exchange information, and therefore, continuously act based on local knowledge, which lead to suboptimal coordination.

4.2 PoC HMT Design

The task allows studying all main teamwork function classes including situation awareness, explainable AI, and working agreements. Four HMT-modules were developed:

1. The *situation awareness* module provides the user with real-time awareness of the positions of the UAVs and of the sensor objects. Via this module the user can intervene by manually identifying sensor objects.
2. The *performance tracking* module provides insight in the identification performance of each UAV. This is a typical XAI function.
3. The *grouping module*. This module allows the user to define groups of UAVs based on their attributes. For example, a group *North* can be defined as being in the upper part of the area, or groups can be defined for better or worse performing UAVs.
4. The *working agreements* module. The user can set up working agreements. Each working agreement consists of three parts: the group that the agreement applies to, whether it concerns permission or prohibition, and third, the action.

The combination of the four modules allows new team coordination possibilities between the human user and the swarm of UAVs. For example, a user is able to define a group *North* and set up a working agreement that this group is prohibited to engage. Although the UAVs themselves have no concept of groups or geographic areas or working agreements, the modules provides an interface for the user to coordinate with the UAVs using these concepts. The grouping, performance tracking and working agreements modules provide an example of the interplay between HMT-modules: several may be combined to obtain a certain functionality. At the same time they are implemented separately, and are reusable for other domains or problems as well.

4.3 PoC Implementation

In order to be able to function in a human-agent team, the UAV is linked to the SAIL framework. The SAIL framework contains an initial version of HATCL that specifies the HATCL message format and domain ontology in plain old java (pojo) classes. All communication between HMT modules and the UAVs follows this ontology.

An interface with semantic anchors is defined that make the UAV interpret and produce HATCL messages. The following information elements are exchanged:

- *Attributes*: ID, location and direction. These are published continuously by the UAV
- *SensorObjects*: sensor objects within the UAV's sensor range are published continuously, including their identification
- *Actions*: the states of the behavior model refer to actions in the HATCL ontology

The interface defines how the UAV responds to HATCL messages. This is defined in the following way:

- *Prohibition*: a prohibition of an action blocks transitions to the corresponding state
- *Permission*: a permission of an action allows transitions to the corresponding state
- *Obligation*: an obligation enforces the transition to the corresponding state
- *Inform*: an inform message with the identification of a contact overrules the internal identification of the UAV

With these HATCL semantic anchors, the UAV can be connected to SAIL. Therewith the UAV can be integrated in a HMT, and will be equipped with social capabilities that allow team collaboration. The elements of communication are added in the domain ontology that is loaded in SAIL. The HMT-modules are implemented as independent scripts that respond to input messages in HATCL (Fig. 5).

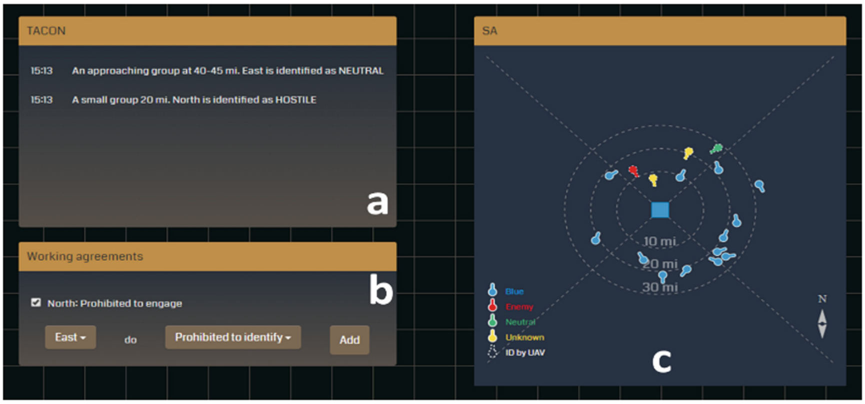


Fig. 5. The GUI of the task environment in the mission phase with external information messages (a) and with the HMT-modules: working agreements (b) and situation awareness (c)

4.4 PoC Evaluation

We set up a pilot experiment to evaluate different HMT concepts in the PoC task. Therefore, we have asked two naïve participants to perform the PoC task with four conditions: two levels of crowdedness of the scenario (many vs. few active UAVs in the task environment) and two HMT-configurations (sparse vs. rich HMT). In the sparse HMT condition they operated with only the *situation awareness* module between them and the UAVs. In the rich HMT condition they had all modules available, and therefore much richer interaction possibilities. The participants were naïve to the task environment and our goals. We measured task scores, specifically the number of hostile actors killed, neutral actors killed (collateral damage), and hits on the main vessel. Also, we asked the participants feedback using a semi-structured interview.

We evaluated the participants performance for the distinct HMT configurations and over multiple runs. Where an improved performance with a rich HMT was expected, this was not the case. First of all, the addition of more HMT modules resulted in worse instead of better performance (see Table 1). Also, the performance improvement with subsequent runs was negligible on all dependent measures (data not presented). Importantly, reduced performance was likely due to imperfections in the implementation of our working agreements module which resulted in unexpected behavior.

Table 1. Average scores in the different experiment settings. Each participant performed three runs of the first three conditions, and two runs of the fourth condition

	Enemies hit	Neutrals hit	Vessel hits
1 Crowded, sparse HMT	59	7	5
2 Crowded, rich HMT	38	4	13
3 Not-crowded, sparse HMT	27	7	2
4 Not-crowded, rich HMT	21	8	6

The results from the participant feedback were very insightful. A first insight is that participants reported that the PoC task and scenario were realistic and sufficiently challenging. The participants did not feel completely in control but partially dependent on the autonomous system, which is exactly the type of HMT situation we wanted to study. Furthermore, it offers broad opportunities for future extension, enabling the development and study of more sophisticated HMT-concepts like trust calibration.

Secondly, participants reported that they wished that the GUI was more refined, for example, showing UAV performance during runtime. Third, participants noted that, on the one hand, they would prefer to have more control over the behavior of the UAVs and that, on the other hand, their cognitive task load was high. This is an interesting tension field in designing an HMT-layer: providing more control options will likely increase task load even more. Fourth, participants noted that they were unsatisfied with system performance when certain working agreements were activated. Specifically, when a group of UAVs was prohibited to identify, the other UAVs did not take this into account when locating themselves. This is an example of a need for system explanation; the user expected more than the system was capable of.

The overall goal of the pilot experiment with the PoC task was to evaluate the usability of the SAIL framework and method to develop and experiment with HMT concepts. We conclude it has shown to be successful for this purpose.

5 Conclusion

We have presented SAIL, a design method and framework for the development of HMT-concepts. The SAIL method integrates human factors and autonomous systems aspects in one design method that is supported by a software framework for implementing and testing HMT-concepts. Starting point of SAIL is that an HMT can be developed without changing the internal capabilities of the autonomous system. The method consists of a iterative development cycle of four steps: problem analysis, HMT-design, prototype implementation and evaluation.

The SAIL framework consists of a modular social layer between autonomous systems and human team members, in which all social capabilities can be implemented that enable teamwork. We have argued that these social capabilities are to some extent generic. Examples are functions for situation awareness, human awareness, explainable AI, working agreements and tasking. Within SAIL, HMT-modules are developed that construct these social capabilities. The modules are reusable in multiple domains.

We have evaluated the SAIL method and framework using a proof of concept task environment. In general we conclude that more complex HMT-systems, the iterative development and evaluation cycle requires a careful approach, as imperfections in the system in early cycles make it hard to properly evaluate HMT-concepts. Nevertheless, our simplified, rudimentary task environment was sufficiently challenging to set up and evaluate a first HMT-concept. It led to valuable feedback and opened up opportunities for more sophisticated HMT-concepts, informing design and implementation early in the development process. Therefore, we believe that the SAIL method and framework have shown to be a promising approach to design, implement and evaluate HMT-concepts. Using SAIL, social requirements for autonomous systems may be developed, that in the end can be adopted in the development of autonomous systems.

References

1. Vinciarelli, A., Pantic, M., Heylen, D., Pelachaud, C., Poggi, I., D'Errico, F., Schroeder, M.: Bridging the gap between social animal and unsocial machine: a survey of social signal processing. IEEE Trans. Affect. Comput. **3**(1), 69–87 (2012)
2. Roff, H., Moyes, R.: Meaningful Human Control, Artificial Intelligence and Autonomous Weapons. In: Briefing Paper Prepared for the Informal Meeting of Experts on Lethal Autonomous Weapons Systems, UN Convention on Conventional Weapons, April 2016
3. Parasuraman, R., Barnes, M., Cosenzo, K., Mulgund, S.: Adaptive automation for human-robot teaming in future command and control systems. Int. J. Command Control **1** (2), 43–68 (2007)
4. van Diggelen, J., Looije, R., van der Waa, J., Neerincx, M.: Human robot team development: an operational and technical perspective. In: Chen, J. (ed.) Advances in Human Factors in Robots and Unmanned Systems. AISC, vol. 595, pp. 293–302. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-60384-1_28
5. Johnson, M., Bradshaw, J.M., Feltovich, P.J., Van Riemsdijk, M.B., Jonker, C.M., Sierhuis, M.: Coactive design: designing support for interdependence in joint activity. J. Hum.-Robot Interact. **3**(1), 2014 (2014)

6. Langley, P., Meadows, B., Sridharan, M., Choi, D.: Explainable agency for intelligent autonomous systems. In: AAAI, pp. 4762–4764, February 2017
7. Lee, J.D., See, K.A.: Trust in automation: designing for appropriate reliance. Hum. Factors **46**(1), 50–80 (2004)
8. Endsley, M.R.: Toward a theory of situation awareness in dynamic systems. Hum. Factors **37**(1), 32–64 (1995)
9. Uszok, A., Bradshaw, J., Jeffers, R., Suri, N., Hayes, P., Breedy, M., Bunch, L., Johnson, M., Kulkarni, S., Lott, J.: KAoS policy and domain services: toward a description-logic approach to policy representation, deconfliction, and enforcement. In: 2003 Proceedings of the IEEE 4th International Workshop on Policies for Distributed Systems and Networks, POLICY 2003, pp. 93–96. IEEE (2003)
10. Dignum, M.V.: A model for organizational interaction: based on agents, founded in logic. SIKS (2004)
11. Fortino, G., Garro, A., Russo, W.: An integrated approach for the development and validation of multi-agent systems. Int. J. Comput. Syst. Sci. Eng. **20**(4), 259–271 (2005)
12. Hayes, B., Shah, J.A.: Improving robot controller transparency through autonomous policy explanation. In: Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction, pp. 303–312. ACM, March 2017
13. Van Lent, M., Fisher, W., Mancuso, M.: An explainable artificial intelligence system for small-unit tactical behavior. In: 2004 Proceedings of the National Conference on Artificial Intelligence, pp. 900–907. AAAI Press, July 2004
14. Labrou, Y., Finin, T., Peng, Y.: Agent communication languages: The current landscape. IEEE Intell. Syst. Their Appl. **14**(2), 45–52 (1999)
15. FIPA: FIPA ACL message structure specification. Foundation for Intelligent Physical Agents (2002). http://www.fipa.org/specs/fipa00061/index.html. Accessed December 2017
16. Peeters, M.M., Bosch, K.V.D., Neerincx, M.A., Meyer, J.J.C.: An ontology for automated scenario–based training. Int. J. Technol. Enhanc. Learn. **6**(3), 195–211 (2014)

Agent-Based Model of Smart Social Networking-Driven Recommendations System for Internet of Vehicles

Kashif Zia[1], Arshad Muhammad[1(✉)], Dinesh Kumar Saini[1], and Alois Ferscha[2]

[1] Faculty of Computing and Information Technology, Sohar University, Sohar, Oman
{kzia,amuhammad,dinesh}@soharuni.edu.om
[2] Johannes Kepler University, Linz, Austria
ferscha@soft.uni-linz.ac.at

Abstract. Social aspects of connectivity and information dispersion are often ignored while weighing the potential of Internet of Things (IoT). Assuming a more commonly acceptable standardization of Big Data generated by Internet of Vehicles (IoV), the social dimensions enabling its fruitful usage; emerging as Social IoV (SIoV); remains a challenge. In this paper, an agent-based model of information sharing (for context-based recommendations) of a hypothetical population of smart vehicles is presented. Some important hypotheses are tested under reasonable connectivity and data constraints. The simulation results reveal that closure of social ties and its timing impacts dispersion of novel information (necessary for a recommender system) substantially. It is also observed that as the network evolves as a result of incremental interactions, recommendations guaranteeing a fair distribution of vehicles across equally good competitors is not possible.

Keywords: Internet of Things · Social IoV · Agent-based model
Context-awareness

1 Introduction

The latest manifestation of "all connected world" is Internet of Things (IoT) [14]. At the physical level, objects in IoT have embedded processors and capability to communicate among them using wired or wireless connections [3,11]. According to the recent predictions by CISCO, 50 billion "things" will be connect to the Internet by 2020 [8]. IoV [6,10] is more than numerous sensors embedded in modern vehicles, which can not only receive the information from its surroundings, but also transmit information, assisting in navigation and traffic management [2]. A practical realization of vehicular technologies is Vehicular Ad hoc NETworks (VANETs) [5], which resulted in important applications regarding handling traffic and better driver/passengers experience [2]. On the other side, we have Vehicular Social Networks (VSN) where passengers can exchange

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 275–287, 2018.
https://doi.org/10.1007/978-3-319-94580-4_22

data related to entertainment, social networks and situations [1] However, IoV is considered more than this – whereby, vehicles build and manage their own social network – more appropriately termed as Social Internet of Vehicles (SIoV).

In general, SIoV is a social network of vehicles within themselves, still, for practical purposes, it cannot be separated from the drivers or owners of these vehicles. Hence, when a vehicle is a part of SIoV, it must build and use the network to achieve owners' goals. Although, it can be debated, but the properties of human social network can be mapped onto buildup and evolution of inter-vehicle social network. The interfacing of the network, and the extend of user (owner of vehicle) satisfaction with respect to goal achievement is a gray area for the purpose of this paper, but it is possible using vehicular technologies.

This paper investigates the potential of SIoV by designing an agent-based model and simulating it in various conditions. The model's conceptualization is based on basic assumptions about IoT connectivity, principles of social network evolution, and users profiling. User profiles and activities serve as an underpinning for the model. Some interesting what-if questions are asked against a couple of intuitive hypotheses. However, the study revolves around finding the extend to which SIoV is capable of providing correct recommendations to their owners in the wake of changing dynamics of the resources.

The rest of the paper is organized as follows. In Sect. 2, we present the scenario, and research hypotheses adapted from the scenario. The proposed model is explained in Sect. 3. Simulation settings, and analysis of the simulation results is presented in Sect. 4 followed by Sect. 5 provides discussion about the hypothesis and conclusion.

2 Scenario and Research Hypothesis

We usually spend most of our time during the weekdays to commute between work, home, school, and shopping. Drivers use number of applications such as navigators [12] to reduce the travel time or to choose between different available options. Moreover, in recent years, Location Based Social Networks (LBSN) [4], i.e Facebook Places, Google maps etc. have gained popularity, where users can share their physical locations, experiences, and ratings etc. Due to these developments, recommender or recommendation systems have gained popularity in recent years where big data is driving force to provide context-aware recommendations to their users.

Realization of a recommendation system enabled by the capabilities of SIoV is an exciting topic. It becomes achievable due to recent industrial endeavors. For example, In CarStream project [15], a technological integration is performed to provide data-driven services. Over 30,000 chauffeured driven cars are connected, the system collects a variety of data such as "vehicle status, driver activity, and passenger-trip information". The user demand is generated by combining parameters such as pickup point, pickup time, arrive time and destination. Motivated from these data fields, our model scenario utilizes the concepts of *ID*, *Time* and *Duration*, to describe a Point of Interest (PoI) from vehicle or user perspective, where *ID* is the identity of the destination, *Time* is the time to reach the destination, and *Duration* is the time to stay at the destination.

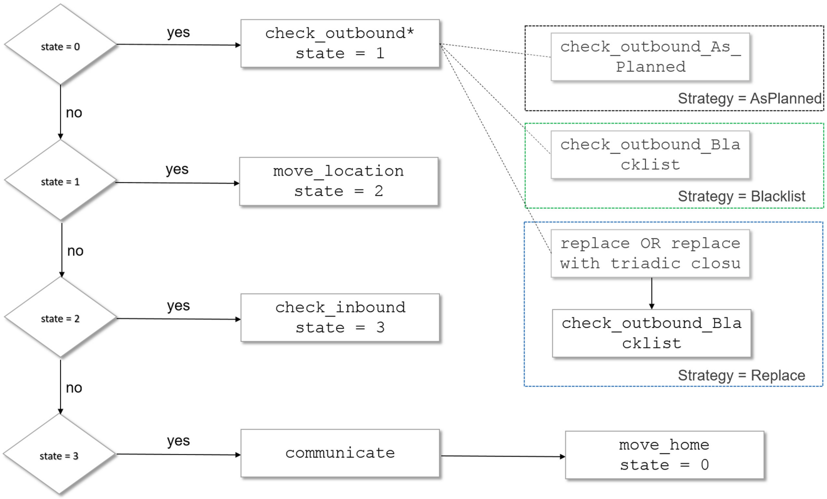


Fig. 1. Vehicle state transitions to be performed at each iteration (= one hour) of the simulation by all the vehicles.

User profiling (motivated from [9]) is used to generate a *plan* in which a user has to visit some PoIs on a weekly basis. Each user has an *expectation*, which is a static personal trait. If the *quality* of service provided by a PoI at the time of a visit is not up to the user's expectation (user *experience*), the PoI is considered as *suspended* for that user. Vehicles connect with each other if they are at the same PoI, thus evolving a social network of vehicles with repeated encounters. The question asked in this paper is: to what extent SIoV is capable of providing the required PoIs recommendations to the users if a user has most of PoIs suspended due to poor quality and/or high expectation.

Before, stating the exact hypotheses, it is worth noting that the vehicles are not different from users. The interface between a vehicle and its owner (user) is assumed to be seamless so that a vehicle know about its owner's plan and also if the current experience about a PoI was good or bad. Vehicle is also capable of changing owner's plan without owner approval. To avoid unnecessary complexity, all these simplifications are made to keep our focus on the evaluation of the following research hypotheses and an agent-based model and simulation is used to verify these hypotheses:

– *Hypothesis 1:* Since connectivity potentially transforms into interaction (information sharing), an increased connectivity degree provides more prospect of having novel information; in short, a *novelty in information is supported by triadic closure* [7].
– *Hypothesis 2:* Since information sharing should increase the user experience, early the information sharing starts, the better would be the user experience; in short, *novel information sharing is directly proportional to how early the information sharing starts.*

– *Hypothesis 3:* Due to intrinsic random nature of individual plans, *a fair distribution of resources should be guaranteed as a result of novel information sharing.*

3 Model

The purpose of the vehicles is to visit PoIs given in the plan at their prescribed time. The three strategies adopted for this model are: (i) AsPlanned, (ii) Blacklist, and (iii) Replace (without and with triadic closure). Section 3.1 discussed the model flow, followed by Sect. 3.2 to discuss the different strategies used in the model.

3.1 Model Flow

Vehicle/User Plan. Plan is a matrix of five columns. The first column contains the *ID* of a PoI, second column is the *time* to visit that PoI. The third column is the *duration* of the visit. The fourth column stores numeric representation of last visit *experience* of the user (vehicle) about that PoI. The fifth column represents if this PoI is *suspended* or not. A list of random PoIs are chosen along with corresponding visit times and visit duration, which are also random. One simulation iteration is equal to one hour. Initially, a random weekly plan, constituted by three components (scheduled visits) is generated, which executes on weekly basis.

Vehicle States. The transition system of vehicles' behavior is divided into five states. If $state = 0$, the vehicle is located at user's home. A vehicle will transit to the next state ($state = 1$) if it is *time* to execute a component of the plan. This is done through `check_outbound*` procedure. If $state = 1$, the vehicle moves to the desired location along with transiting to the next state ($state = 2$). This is done through `move_location` procedure. If $state = 2$, and the *duration* to stay at the current location is expired, the vehicle will transit to the next state ($state = 3$). This is done through `check_inbound` procedure. If $state = 3$, the vehicle will communicate with its neighbors (in proximity). This is done through `communicate` procedure. After communicating, the vehicle will move back to home and resets its state to 0. This is done through `move_home` procedure. Figure 1 presents a graphical view of vehicles' state transitions. In the following, a description of each of these procedures is given.

Table 1. Plan matrix initialization (vehicle 569)

ID	Time	Duration	Experience	Suspended?
1059	17	2	1	False
1054	122	4	1	False
1053	50	2	1	False

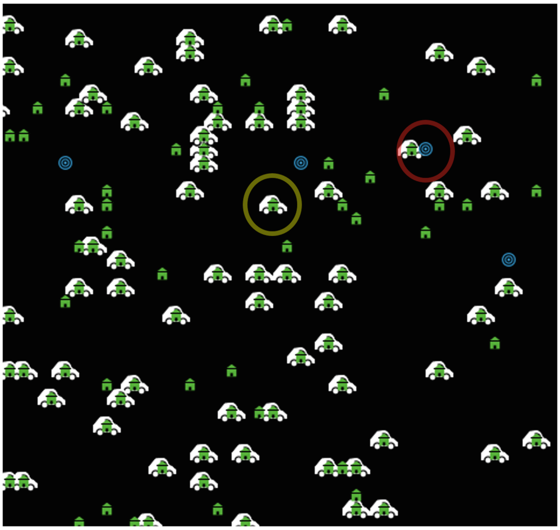


Fig. 2. A view of one fourth of the simulation world. Vehicle with yellow circle is vehicle 569, and PoI with red circle is PoI 1059. (Color figure online)

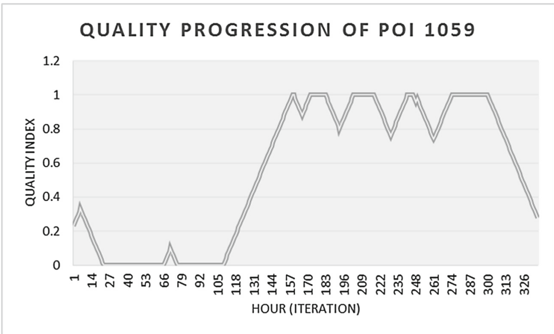


Fig. 3. Quality progression of PoI 1059 of first two weeks.

check_outbound.* The plan matrix contains three components (scheduled visits), which are randomly set at the start of the simulation. Table 1 shows an example of this matrix for a sample vehicle, with ID 569. At hour 17, this vehicle has to visit PoI 1059 for a duration equal to 2 h. The `check_outbound*` procedure, at iteration 17 would identify it and the state of the vehicle would change from 0 to 1. Similarly, at their turn, PoI 1054 and 1053 will also be visited. By default, the experience of the user regarding the recent visit to a PoI is set to default positive extreme equal to 1. As shown in Fig. 1, the procedure `check_outbound*` has three varieties which correspond to three different strategies of selection of a PoI. Here, we have explained strategy of selecting a PoI according to the initial plan, named as `check_outbound_As_Planned` (replacing * by the strategy used). Obviously, this strategy is static, just executing the plan as it is.

move_location. A vehicle moves to the PoI selected in `check_outbound*` procedure. The current *quality* offered by the PoI with a random variation of ±25 % is stored in the plan matrix as experience of the user. For example, vehicle 569 has to visit PoI 1059 at iteration 17 (also see Fig. 2). The progression curve of *quality* offered by PoI 1059 is shown in Fig. 3. At iteration 17, the value is 0.092 which is transformed into *experience* of the vehicle and stored into plan matrix as 0.0909 (see updated matrix of Table 2).
check_inbound. If visit duration of a vehicle at a PoI is exhausted, the state of the vehicle transits into 3. After completing 2 units of duration at PoI 1059, vehicle 569's state changes to 3 at iteration 20.
communicate. The vehicles co-located at the same PoI network with each other to form ties. Initially, it is a weak tie. With repeated encounters, a weak tie becomes a strong tie. Each vehicle has a data structure, which contains this information. For each vehicle which has formed a tie with another vehicle, the array of information is constituted by a tuple "time of the latest encounter, how many encounters?, strong tie?". If number of encounters with the same vehicle reaches to a *threshold*, a weak tie changes into a strong tie.

For example, vehicle 569 at iteration 20 encounters two more vehicles at PoI 1059. It is evident from Table 3, both these vehicles are encountered for the first time (how many encounters? = 1) and have established a weak tie with vehicle 269 (strong tie? = false). With repeated encounters at same or other locations, a weak tie will change into a strong tie, if the number of encounters are equal to the *threshold*. It is worth noting that communication is not used in first two strategies. Hence, in first two strategies, after completion of duration to stay at a PoI, a vehicle moves back to its home.
move_home. A vehicle moves back to its home, resetting its state from 3 to 0.

Table 2. Updated plan matrix (vehicle 569).

ID	Time	Duration	Experience	Suspended?
1059	17	2	0.0909	False
1054	122	4	1	False
1053	50	2	1	False

Table 3. Vehicles data (vehicle 569).

Vehicles data of vehicle 569		
984	Time of latest encounter	20
	How many encounters?	1
	Strong tie?	False
691	Time of latest encounter	20
	How many encounters?	1
	Strong tie?	False

3.2 Strategies

AsPlanned implemented through `check_outbound_AsPlanned` as explained in the last subsection.

Blacklist implemented through `check_outbound_Blacklist`. From a vehicle's plan, the method of selection of a visits works as follows. If the vehicle has to visit a PoI at the current hour and that PoI is not already blacklisted (if blacklisted then do not visit any location), then the quality of current PoI will be evaluated against vehicle's *expectation*. If quality is greater than expectation, the vehicle will transit to state 1, ready to visit the current PoI. Otherwise, the PoI will be suspended in the plan matrix. For example, at the 17th hour of the second week, the last experience of vehicle 569 about PoI 1059 was 0.0909, which is well below its expectation (0.465640). Hence, PoI 1059 will be suspended in vehicle 569 plan (see Table 4). Further, an alternate PoI would be sought that is not yet blacklisted to be executed in the current hour. If no such PoI is found, the vehicle would not visit any location (state remains 0).

Table 4. Plan matrix update at 17th hour of second week (vehicle 569).

ID	Time	Duration	Experience	Suspended?
1059	17	2	0.0909	True
1054	122	4	0.5684	False
1053	50	2	0.3325	False

Replace implemented through `check_outbound_replace`. In this procedure, all those PoIs that have been suspended would be replaced by other PoIs which are not suspended. The information about these PoIs would come from "friends" (having strong ties) of a vehicle. The whole suspended row in plan matrix of the vehicle would be replaced by a not-suspended row attained from a friend.

After replacement, `check_outbound_Blacklist` procedure would be called to select a PoI. Replace has two varieties:

- *Replacement without triadic closure:* Replacement of blacklisted PoIs can be performed without considering the internal social networking dynamics, as mentioned above.
- *Replacement with triadic closure:* Replacement of blacklisted PoIs can also be performed while considering the internal social networking dynamics. One of the most basic and most important dynamics of social networks evolution is "triadic closure". The triadic closure property states that, if a node a in a social network has strong ties with two nodes b and c, then, there is a likelihood that b and c would becomes friends themselves some times in the future.

4 Simulation and Results

The model is created in NetLogo [13], a multi-agent programming environment. A grid of cells of size 75 × 75 is used to populate agents. There are three types of agents, two static and one mobile. The static PoI and Home agents and randomly placed in the environment. The count of PoIs is 15, available for 500 homes. Vehicles are mobile agents which are 10% more than homes; thus, ensuring attachment of more than one vehicles to a home. A vehicles are attached randomly to a home. The users (owners of the vehicles) are abstract entities encapsulated within vehicles.

The three strategies adopted (as explained before) are evaluated based on three parameters:

- quality-index: When a vehicle visits a PoI, it experiences the quality offered by that PoI at that time. In one week, a vehicle has to visit three PoIs, that is why, the quality-index is the average of three experiences of the vehicle at each hour. The quality-index of the whole simulation space is then an aggregation of all these experiences divided by vehicles count.
- connectivity-index: When two vehicles are at some PoI at the same time, they connect with each other, irrespective of type of tie they may have (weak or strong). The connectivity-index of the whole simulation space is an aggregation of all current (at this hour) connections divided by vehicles count.
- PoI-utilization: The population of vehicles at a PoI at an hour determines its current utilization. Derived from it is the standard deviation of utilization of all PoIs.

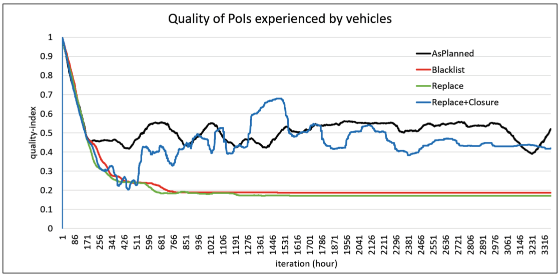


Fig. 4. Comparison of strategies ($th_{strong-tie} = 5$): average quality of PoIs experienced by vehicles averaged across 100 simulation runs.

The graph shown in Fig. 4 presents time series of average quality of PoIs experienced by all the vehicles for a span equal to twenty weeks and averaged across 100 simulation runs. A comparison between four strategies reveals the following:

- Blacklist: Once a PoI is blacklisted, it is not visited by a vehicle anymore. Less number of visits are performed hence reducing the average quality-index, due to the reason that after the first week, only those PoIs are visited which satisfy users/vehicles expectation.
- AsPlanned: Although, the above strategy (Blacklist) presents an extreme case, it can be argued to be the case. But, continuously visiting the same PoIs, which is not up to user expectations is less likely to happen. However, due to randomness in quality parameter of PoIs, the PoIs that did not perform according to the expectations previously, may perform well this time. This is what exactly is revealed in the simulation results, in which, after the first week, quality-index fluctuates around the middle (0.5).
- Replace: The replace strategy replaces a blacklisted PoI that a vehicle wants to visit by a PoI recommended by a friend. Since a friend is a strong tie, the value of threshold (repeated encounters) is important. A higher value of threshold ($th_{strong-tie} = 5$), such as 5 in this case, would delay replacement process. In this case, the process is so delayed that no vehicle has new information for others as all PoIs are already blacklisted.
- Replace with Closure: The triadic closure provides novel information about uncommon PoIs. That is the reason, most of the time, this strategy competes with the AsPlanned strategy.

The graph shown in Fig. 5 verifies the above findings relating the quality index with the average degree of vehicles. The connectivity of strategy replace with closure is by far higher than the other three strategies.

With decrease in encounter threshold of strong tie ($th_{strong-tie} = 2$), the Replace strategy starts competing with Replace strategy with triadic closure. It happens in patches through, where connectivity generally increases jumping up from one phase to another, consequently, increasing the quality index in an incremental way as well (See Figs. 6 and 7). Overall, in terms of quality index replace and replace with triadic closure for $th_{strong-tie} = 2$ works much better than $th_{strong-tie} = 5$.

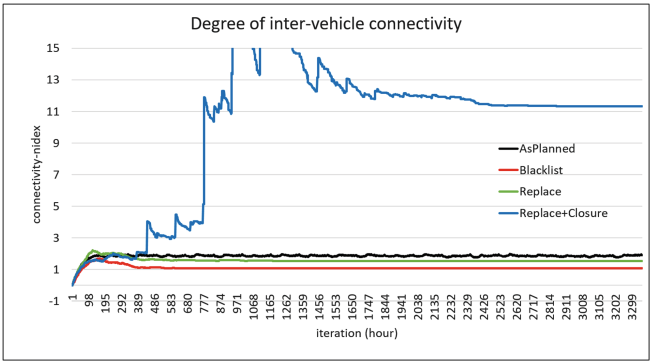


Fig. 5. Comparison of strategies ($th_{strong-tie} = 5$): average connectivity of vehicles averaged across 100 simulation runs.

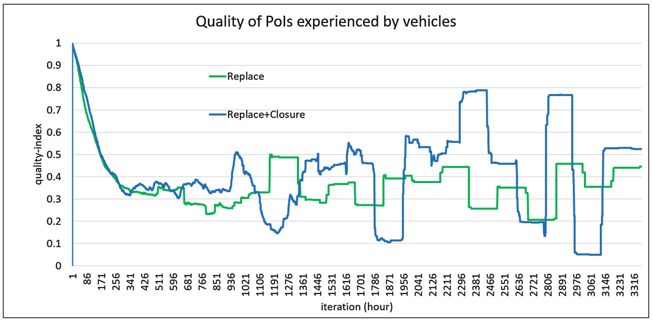


Fig. 6. Comparison of strategies ($th_{strong-tie} = 2$): average quality of PoIs experienced by vehicles averaged across 100 simulation runs.

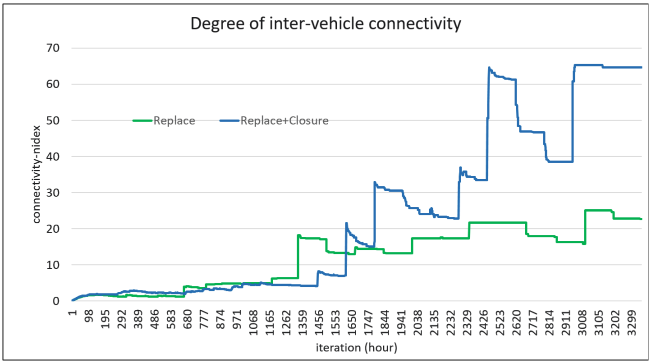


Fig. 7. Comparison of strategies ($th_{strong-tie} = 2$): average connectivity of vehicles averaged across 100 simulation runs.

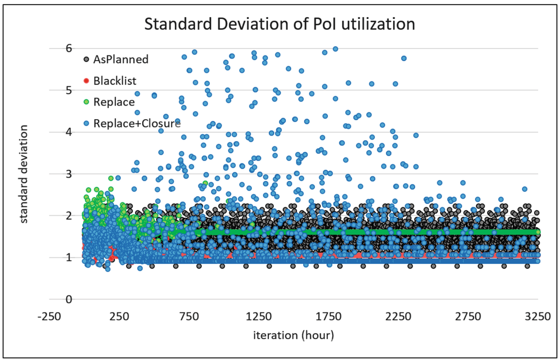


Fig. 8. Comparison of strategies ($th_{strong-tie} = 5$): average standard deviation of PoIs utilization across 100 simulation runs.

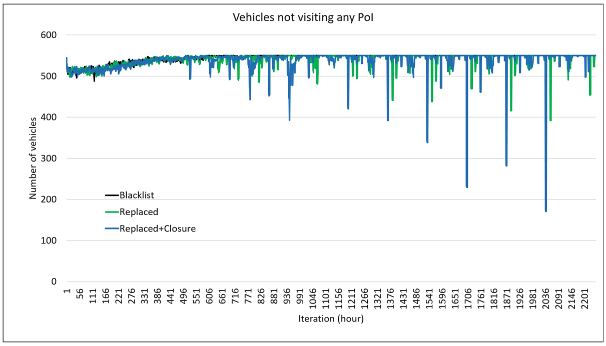


Fig. 9. Comparison of strategies ($th_{strong-tie} = 5$): number of vehicles not visiting any PoI (one simulation run per strategy).

Another exciting aspect that is studies is utilization of PoIs. Although there is no safeguard against it in the model itself, it is undesirable to have high asymmetry in PoIs utilization, which is achieved through standard deviation of PoI utilization (SDU). The graph shown in Fig. 8 compares four strategies. It can be seen that SDU of strategy blacklist is least because there are no visits after week 1, evidenced by the graph shown in Fig. 9. The strategy replace also has a low SDU after first few weeks, but this is also due to no visits. The strategy replace with closure has an erratic behavior, sometimes high and sometimes low. But contrary to the above, it is evident (from Fig. 9) that low SDU is not due to no visits, but, due to the symmetric utilization of PoIs which are not blacklisted. No doubt, high SDU is due to extreme asymmetry in PoIs utilization.

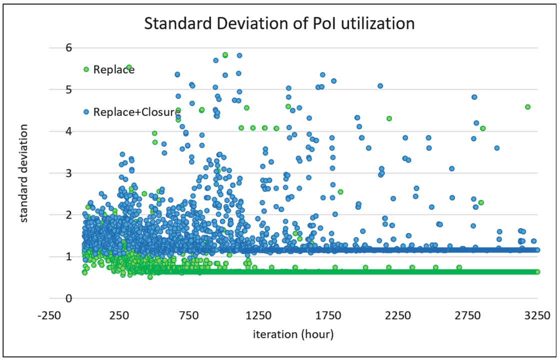


Fig. 10. Comparison of strategies ($th_{strong-tie} = 2$): average standard deviation of PoIs utilization across 100 simulation runs.

The inactivity of replace strategy is compensated by reducing from $th_{strong-tie} = 5$ to $th_{strong-tie} = 2$. Compare strategy replace with strategy replace with closure in the graph shown in Fig. 10. Overall, again, in terms of PoIs utilization, replace and replace with triadic closure for $th_{strong-tie} = 2$ works much better than $th_{strong-tie} = 5$.

5 Discussion and Conclusion

In a dynamically changing quality of service provisioning by PoIs, the vehicles with blacklisted PoIs in their plan can utilize their strong ties to have novel information about some PoIs those are not blacklisted at a particular time. Increase in connectivity is responsible of increasing the quality index as depicted in Figs. 4 and 5. Same is true when threshold is decreased from $th_{strong-tie} = 5$ to $th_{strong-tie} = 2$ (see graphs in Figs. 6 and 7). These results verify *Hypothesis 1*. However, a delay in the information sharing, particularly without triadic closure would not help. This is evident from inability of replace strategy to perform well if $th_{strong-tie} = 5$, whereas, it performs reasonably well if $th_{strong-tie} = 2$. This verifies *Hypothesis 2*. In combination, the following can be concluded:
"Novelty in information is directly proportional to how early the information sharing starts, however, triadic closure is capable of producing novel information even if information sharing starts late."

A cross comparison among Figs. 8 and 10 reveals that for lesser threshold, SDU is even less. Moreover, at $th_{strong-tie} = 5$, replace with closure works better at the early stages of the simulation when compared with replace. This is opposite in case of $th_{strong-tie} = 2$, replace works better at the early stages of the simulation when compared with replace with closure. Overall, SDU is less in the early stages of the simulation and then it becomes erratic, fluctuating between extremely high values to extremely low values. Therefore, *Hypothesis 3* is conditionally verified and can be stated as: **"Fair distribution of PoIs is guaranteed at the start but not later."** In conclusion, closure of social ties and its timing plays an important role in dispersion of novel information. As the network evolves as a result of incremental interactions, recommendations guaranteeing a fair distribution of vehicles across equally good competitors is not possible, particularly at the later stages of interactions.

References

1. Abbani, N., Jomaa, M., Tarhini, T., Artail, H., El-Hajj, W.: Managing social networks in vehicular networks using trust rules. In: 2011 IEEE Symposium on Wireless Technology and Applications (ISWTA), pp. 168–173. IEEE (2011)
2. Alam, K.M., Saini, M., El Saddik, A.: Toward social internet of vehicles: concept, architecture, and applications. IEEE Access **3**, 343–357 (2015)
3. Bakıcı, T., Almirall, E., Wareham, J.: A smart city initiative: the case of Barcelona. J. Knowl. Econ. **4**(2), 135–148 (2013)
4. Bao, J., Zheng, Y., Wilkie, D., Mokbel, M.: Recommendations in location-based social networks: a survey. GeoInformatica **19**(3), 525–565 (2015)

5. Cunha, F., Villas, L., Boukerche, A., Maia, G., Viana, A., Mini, R.A., Loureiro, A.A.: Data communication in VANETs: protocols, applications and challenges. Ad Hoc Netw. **44**, 90–103 (2016)
6. Dandala, T., Krishnamurthy, V., Alwan, R.: Internet of Vehicles (IoV) for traffic management. In: 2017 International Conference on Computer, Communication and Signal Processing (ICCCSP), pp. 1–4. IEEE (2017)
7. Easley, D., Kleinberg, J.: Networks, Crowds, and Markets: Reasoning About a Highly Connected World. Cambridge University Press, Cambridge (2010)
8. Evans, D.: The Internet of Things: how the next evolution of the internet is changing everything. CISCO White Paper **1**(2011), 1–11 (2011)
9. Fagnant, D.J., Kockelman, K.M.: The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios. Transp. Res. Part C: Emerg. Technol. **40**, 1–13 (2014)
10. Fangchun, Y., Shangguang, W., Jinglin, L., Zhihan, L., Qibo, S.: An overview of internet of vehicles. China Commun. **11**(10), 1–15 (2014)
11. Neirotti, P., De Marco, A., Cagliano, A.C., Mangano, G., Scorrano, F.: Current trends in smart city initiatives: some stylised facts. Cities **38**, 25–36 (2014)
12. Park, E., Kim, K.J.: Driver acceptance of car navigation systems: integration of locational accuracy, processing speed, and service and display quality with technology acceptance model. Pers. Ubiquit. Comput. **18**(3), 503–513 (2014)
13. Tisue, S., Wilensky, U.: NetLogo: a simple environment for modeling complexity. In: International conference on complex systems, Boston, MA, vol. 21, pp. 16–21 (2004)
14. Wortmann, F., Flüchter, K.: Internet of Things. Bus. Inf. Syst. Eng. **57**(3), 221–224 (2015)
15. Zhang, M., Wo, T., Xie, T., Lin, X., Liu, Y.: CarStream: an industrial system of big data processing for internet-of-vehicles. Proc. VLDB Endow. **10**(12), 1766–1777 (2017)

Demo Papers

MAXIM-GPRT: A Simulator of Local Schedulers, Negotiations, and Communication for Multi-Agent Systems in General-Purpose and Real-Time Scenarios

Giuseppe Albanese[3], Davide Calvaresi[1,2](✉), Paolo Sernani[3], Fabien Dubosson[2], Aldo Franco Dragoni[3], and Michael Schumacher[2]

[1] Scuola Superiore Sant'Anna, Pisa, Italy
d.calvaresi@sssup.it
[2] University of Applied Sciences Western Switzerland, Sierre, Switzerland
michael.schumacher@hevs.ch
[3] Università Politecnica delle Marche, Ancona, Italy
{p.sernani,a.f.dragoni}@univpm.it, peppe-a@live.it

Abstract. In safety-critical scenarios, the compliance with strict-timing constraints is mandatory. This demo presents a simulator named MAXIM-GPRT enabling the analysis of the behaviors produced by Multi-Agent Systems (MAS) composed of both General-Purpose (GP) and Real-time (RT) algorithms. Therefore, MAXIM-GPRT is crucial to prove that current MAS cannot provide timing guarantees, nor guarantee correct behaviors in the worst case scenario. However, adopting and adapting models and algorithms from RT systems, such a compliance, can be achieved.

Keywords: MAS · Timing-reliability · Deadline missing-rate
MAS simulator

1 Introduction

In a society increasingly interconnected, the new generation of systems are evolving towards the *Internet of Everything* (IoE). In the IoE era, a multitude of distributed electronic devices, continuously interacting with the environment and collecting data from each other, couple virtual and real domains by reconciling so-called Cyber-Physical System (CPS) solutions. However, CPS are increasingly employed in safety-critical scenarios. Their behavior is therefore required to be correct in terms of both result and deliberation time (i.e., dependable and reliable behaviors). Unfortunately, current frameworks supporting the development of Multi-Agent Systems (MAS) only adopt General-Purpose (GP) algorithms pursuing the *best-effort approaches* [3], which give no means to offer any timing guarantee, not off-line nor on-line. Hence, there is a need for understanding

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 291–295, 2018.
https://doi.org/10.1007/978-3-319-94580-4_23

and evaluating the system behavior in respecting timing constraints and giving timing guarantees according to given inputs and systems setups.

2 Main Purpose

According to Calvaresi et al. [4], current multi-agent platform and applications are incapable of enforcing the compliance with strict timing constraints (impossibility of providing any guarantee about the system behaviors in the worst-case scenario). Therefore, the adoption of MAS is hampered, excluding significant application scenarios such as "safety-critical environments".

The main reasons for this lack of real-time (RT) satisfiability in MAS originate from current theories, standards, and technological implementations. In particular, traditional internal agent schedulers, communication middlewares, and negotiation protocols have been identified as co-factors inhibiting real-time compliance [2].

Table 1. Simulation results

Id	Indicator	Description
I1	Deadline Miss Ratio (DMR)	Number of deadlines missed by a task in a given simulated time
I2	Lateness (LT)	Extra time required by a task missing its deadline to complete
I3	Response Time (RTM)	Amount of time required to complete a given task

Table 2. Configurable parameters

Id	Parameter	Description
P1	Number of agents	Number of agents participating in the simulation
P2	Agent utilization	Load of the agent's CPU (see Sect. 3.1)
P3	Agent knowledge	Set of tasks an agent is able to execute
P4	Agent task-set	Set of running tasks
P5	Agent services	Set of tasks an agent might execute on demand
P6	Agent needs	Set of tasks an agent needs, but it is unable to execute
P7	Tasks models	Typology of running tasks
P8	Tasks utilization	Load of a single task (see Sect. 3.1)
P9	Negotiation prot	Mechanisms used to negotiate task execution
P10	Heuristics	Policies used by agents to select possible contractors and to award them

The main purpose of this demonstration is to present MAXIM-GPRT, a simulator developed to study MAS performances, possible risks, and failures in both GP and RT scenarios. MAXIM-GPRT allows to integrate MAS with the RT theory that, by studying computing systems that must guarantee bounded and predictable response times [1], provides the means to evaluate timing constraints and guarantees. In particular, the schedulers and negotiation protocols used in the most known agent frameworks [3], and a selection of schedulers typical of real-time systems, have been implemented inside the simulator and made modular. Table 1 describes the possible outputs provided by the simulator. Table 2 details the configurable elements characterizing MAXIM-GPRT.

3 Demonstration

MAXIM-GPRT relies on the OMNET++ framework[1], and is composed of an arbitrary amount (P1) of simple modules organized in a fully connected network with configurable channels (e.g., type of connection, communication delays, and connected modules). Such modules embody the agents in a community. They have the same structure with customizable attributes and capabilities (see Fig. 1(a)). An additional agent is by default part of the community. It is the Directory Facilitator (DF). Such a concept is adopted from the FIPA standard for agent management[2] and is in charge of providing a list of agents willing to offer given services. To enable completely autonomous executions and dynamics, the parameters listed in Table 2 have to be defined before starting the simulation. Such parameters (mainly organized in XML files) can be defined by hand or automatically, exploiting a tool to generate them according to given ranges and distributions.

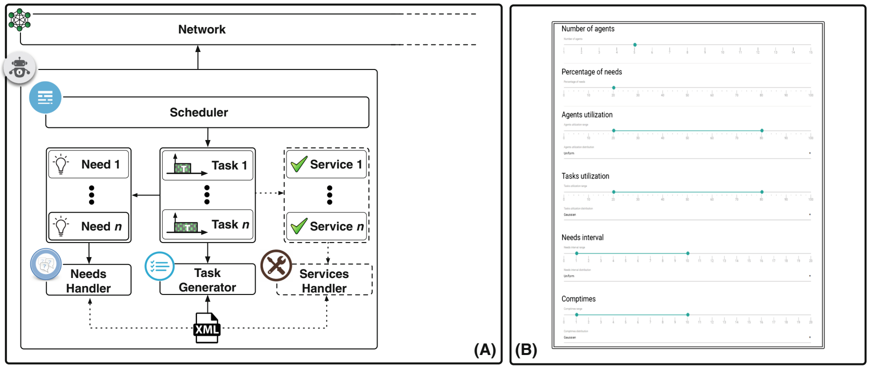


Fig. 1. (a) Agent composition, (b) parameters generator web interface.

[1] https://omnetpp.org/.
[2] http://www.fipa.org/specs/fipa00023/.

3.1 MAXIM-GPRT: Possible Setups

Figure 1(b) shows the two interfaces (command line and web) to setup the parameters in Table 2 with the following values:

[P1]: an integer value ($x \geq 1$); [P2]: a real value ($0 < x \geq 1$); [P3]: a set of tasks characterized by: id, executor, demander, computation time[3], residual computation time, arrival time (see footnote 3), relative deadline, period (see footnote 3), number of executions, first activation time, last activation time, public flag, server id (XML format); [P4]: a sub-set of the tasks (marked as activable) in P3. [P5]: a sub-set of tasks (marked as public) in P3. [P6]: a sub-set of tasks that an agent is unable to execute, and that might be part of someone else P5. Each need is characterized by a starting time and a number indicating its required execution(s); [P7]: periodic, periodic in an interval, aperiodic; [P8]: a real value (see footnote 3) ($0 < x \geq 1$); [P9]: Contract Net Protocol and Reservation-Based Negotiation Protocol; [P10]: (H1) select first agent in the list, (H2) select a random agent, (H3) select a random subset of agents, (H4) select the best offer according to a cost function;

P1 and P9 are the only parameters valid for the entire community. The remaining parameters can be set for every agent singularly.

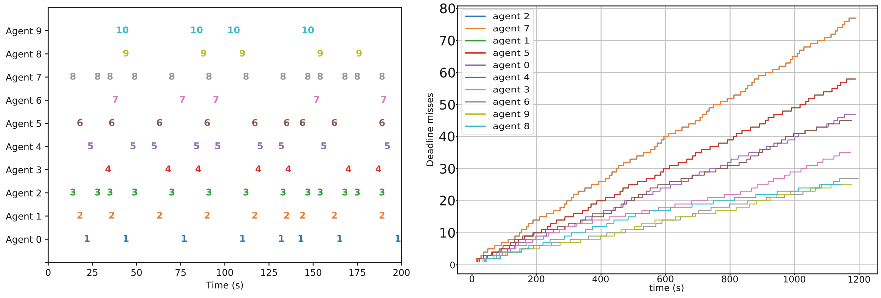
3.2 MAXIM-GPRT: Demostration of Simulation Analysis

The proposed demo shows how simulation analysis of deadline miss and task-set scheduling can be performed on a MAS. Every simulation can be run "event per event", at a given (arbitrarily fast) speed, or completed in "one click". At the completion of every simulation, MAXIM-GPRT provides reports composed of:

DMR: for every occurrence, it contains time and details of the tasks that missed their deadlines, the agent's utilization factor, and the running tasks at that given point in time. *LT:* it integrates the DMR info reporting the extra time needed to complete the execution by tasks missing their deadlines. *RTM:* it provides the response time of all the executed tasks (including those missed their deadlines). *Statistics:* it contains the total number of deadlines checked, and the number of deadlines missed for each agent in the platform during a given simulation.

Concerning the indicators presented in Table 1, we developed a script to plot the following information collected in the simulation log. The demo will show the available graphs that are: *(i)* the plot of given task(s) missing its deadline(s) in a given agent(s) in the total or a given time interval (see Fig. 2a), *(ii)* the plot of the cumulative deadline missed by every agent that took part in a given simulation over the total or a given time interval (see Fig. 2a), *(iii)* plots aggregating data collected by the logs and the reports obtain after several simulations in similar and different setups[4].

[3] Values computed according to a uniform probability distribution.
[4] Due to space limitations only *(i)* and *(ii)* have been shown.



(a) DMR for 10 agents, [0-200s].

(b) DMR for 10 agents, [0-1200s].

Fig. 2. Results of the deadline miss analysis.

4 Conclusions

Pursuing the timing-reliability in MAS, the proposed work provides a tool, MAXIM-GPRT, to analyze deadline miss ratio, response-time, and lateness of the agent's behaviors (see Table 1) employing GP and RT algorithms (see Table 2). In light of the findings produced by MAXIM-GPRT, it can be concluded that to be able to employ MAS in scenarios demanding the compliance with strict-timing constraints, the adoption and adaption of real-time scheduling models is crucial. The development of the MAXIM-GPRT simulator revealed to be strategic to prove that a task missing its deadline can be the output of several factors such as the agent utilization factor, single task utilization factor, and task-set composition. As an ongoing work, we are studying a qualitative evaluation in terms of response time and lateness (in case of deadline miss) between FCFS, RR, and EDF (CBS when employing sporadic tasks).

References

1. Buttazzo, G.: Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications, vol. 24. Springer, Heidelberg (2011). https://doi.org/10.1007/978-1-4614-0676-1
2. Calvaresi, D., Appoggetti, K., Lustrissimini, L., Marinoni, M., Sernani, P., Dragoni, A.F., Schumacher, M.: Multi-agent systems negotiation protocols for cyber-physical systems: results from a systematic literature review. In: Proceedings of 10th International Conference on Agents and Artificial Intelligence (2018)
3. Calvaresi, D., Marinoni, M., Lustrissimini, L., Appoggetti, K., Sernani, P., Dragoni, A.F., Schumacher, M., Buttazzo, G.: Local scheduling in multi-agent systems: getting ready for safety-critical scenarios. In: Proceedings of 15th European Conference on Multi-Agent Systems. Springer (2017)
4. Calvaresi, D., Marinoni, M., Sturm, A., Schumacher, M., Buttazzo, G.: The challenge of real-time multi-agent systems for enabling IoT and CPS. In: Proceedings of the International Conference on Web Intelligence, pp. 356–364 (2017). https://doi.org/10.1145/3106426.3106518

Managing Bad AIPs with RIVERtools

Davide Ancona, Angelo Ferrando, Luca Franceschini, and Viviana Mascardi(✉)

Università degli Studi di Genova, Genoa, Italy
{davide.ancona,angelo.ferrando,luca.franceschini,
viviana.mascardi}@dibris.unige.it

Abstract. We present the RIVERtools integrated development environment for specifying Agent Interaction Protocols (AIPs) modelled as trace expressions, and for statically verifying some of their properties. In particular, this demonstration paper aims at showing why a "good" AIP can become a "bad" one because of unreliability of some communication channels, and how RIVERtools can cope with such bad AIPs, suggesting to the developer possible ways to dynamically verify them in a partially decentralized way.

1 Introduction

In the companion paper of this document [2] we provided a taxonomy of Agent Interaction Protocols (AIPs) based on the possibility to enact them. AIPs usually describe the protocol as a whole, taking a **global** viewpoint, but when moving from the specification to the execution stage, an AIP must be enacted by agents in the MAS. Enacting an AIP requires at least to be able to derive the **local** portion of the AIP each agent is in charge of. In [2] we defined as "good" AIPs those that can be enacted. The following static properties are necessary conditions for enactment:

1. *connectedness for sequence*: there is always an evident cause-effect link between two sequential messages (if a sends a message to b and, as a consequence, c is expected to send a message to d, this property is violated as there is no causal flow between the messages);

2. *unique point of choice*: in case some mutually exclusive choice must be made, the choice is up to only one agent involved in the protocol (protocols should not contain choices like "either a sends a message to b, or c sends a message to d").

We call the points of the AIP where these properties are violated, critical points. "Bad" AIPs are those that do not enjoy the properties above, and hence contain critical points. "Ugly" AIPs are those that do not even respect the AIP specification language grammar, or other hard constraints to be satisfied at design time.

Recognizing bad and ugly AIPs is of paramount importance to design AIPs which meet the system requirements. However, bad AIPs should not be necessarily discarded, as they might be the result of good AIPs where some communication channels demonstrate not to be fully reliable. In particular, although agents

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 296–300, 2018.
https://doi.org/10.1007/978-3-319-94580-4_24

involved in critical points cannot be enacted, partially decentralized runtime verification of bad AIPs can still be carried out on agent subsets [6].

2 Main Purpose

The main purpose of this demonstration paper is to show how RIVERtools [5] deals with bad AIPs. This is an original contribution, not published elsewhere.

RIVERtools is an Eclipse plugin developed using Xtext (www.eclipse.org/Xtext/). Since Xtext supports Xtend (www.eclipse.org/xtend/), a dialect of Java, the development of RIVERtools has been obtained in a fast and maintainable way. RIVERtools supports the definition of AIPs using the trace expressions formalism [1,3,4].

A trace expression τ represents a set of possibly infinite event traces, and is defined on top of the following operators: (1) ϵ (empty trace), denoting the singleton set $\{\epsilon\}$ containing the empty event trace ϵ; (2) $\vartheta{:}\tau$ (*prefix*), denoting the set of all traces whose first event e matches the event type[1] ϑ ($e \in \vartheta$), and the remaining part is a trace of τ; (3) $\tau_1{\cdot}\tau_2$ (*concatenation*), denoting the set of all traces obtained by concatenating the traces of τ_1 with those of τ_2; (4) $\tau_1{\wedge}\tau_2$ (*intersection*), denoting the intersection of the traces of τ_1 and τ_2; (5) $\tau_1{\vee}\tau_2$ (*union*), denoting the union of the traces of τ_1 and τ_2; (6) $\tau_1|\tau_2$ (*shuffle*), denoting the set obtained by shuffling the traces of τ_1 with the traces of τ_2.

3 Demonstration

How to Define AIPs in RIVERtools. The developer must install the Xtext framework and the RIVERtools plugin on his/her own version of Eclipse. After that, he/she can create a new project containing a ".texp" file for each AIP he/she wants to define. The RIVERtools plugin checks the protocol and notifies the developer in case problems (syntax error, but also violation of connectedness for sequence and unique point of choice properties) are detected.

If the AIP is syntactically correct, RIVERtools compiles it into the corresponding SWI-Prolog (www.swi-prolog.org) code that will be used during the MAS runtime verification stage [3,4].

Motivating Example. The protocol presented in Fig. 1 is the simple BookShop AIP introduced in [2]. It involves `alice`, `barbara`, `carol`, `dave`, `emily` and `frank`, defined using the `roles` keyword. The AIP models the protocol `alice` should follow to obtain the book she is interested in. The message exchange can take place via `whatsapp` and via `email`. Starting from the global AIP specification, RIVERtools can extract the local perspectives (or "local viewpoints", or "projections") related to subsets of agents, using the `partition` keyword. Subsets may involve one or more agents. Figure 1, for instance, shows how the

[1] To be more general, trace expressions are built on top of event types (chosen from a set $\mathcal{ET}$), rather than of single events; an event type denotes a subset of $\mathcal{E}$.

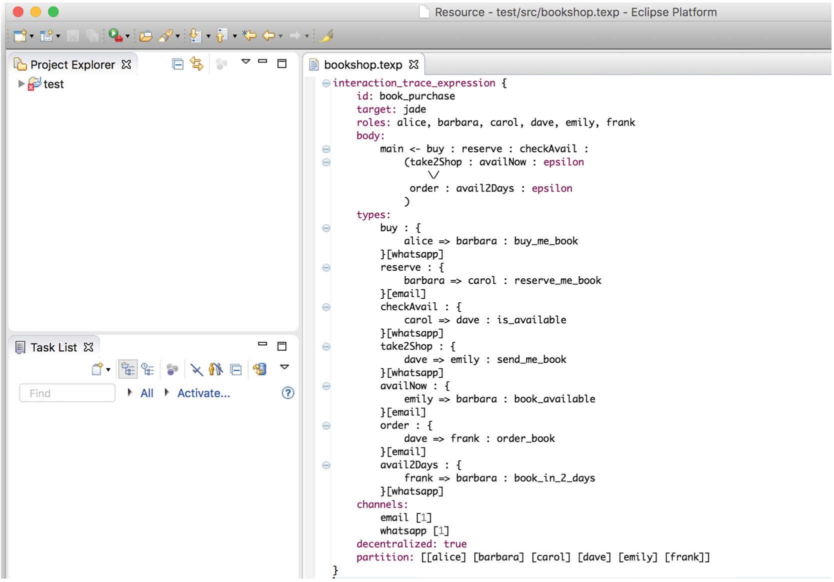
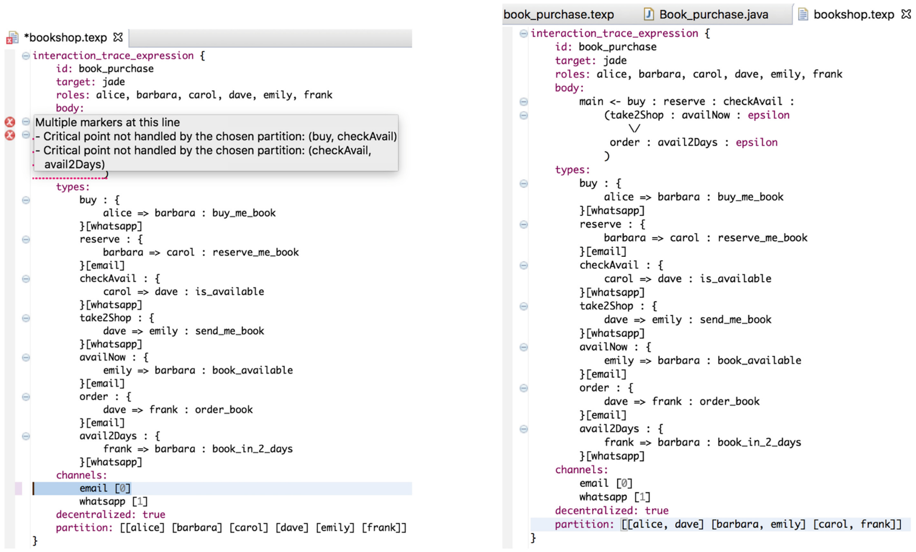


Fig. 1. Book-Shop AIP in RIVERtools.

global Book-Shop AIP can be "projected" onto each single participant, obtaining six different local perspectives. When the protocol is a good one, it is always possible to project the global protocol onto individual agents. This is no longer true when the protocol is bad.

RIVERtools allows the MAS developer to define channels for the communication among agents and to associate each message with the channel it will be sent over. Each channel can have a numeric information associated with, used by RIVERtools to model the channel reliability. As an example, in Fig. 1 both `whatsapp` and `email` channels have reliability `1`, meaning that they are fully reliable. How can bad protocols be generated? Usually, we do not intentionally design bad protocols from scratch, but we obtain them starting from good ones where some channels are unreliable, and a monitor in charge for runtime verification could not be able to observe all the exchanged messages. RIVERtools detects the protocol features and informs the developer: in Fig. 2(a), we can see the RIVERtools message when we change the reliability of the `email` channel from `1` (fully reliable, all messages can be observed) to `0` (fully unreliable, no message can be observed). Since we have set the reliability value of the `email` channel to `0`, the `order_book`, `reserve_me_book`, `book_available` messages exchanged using this channel are not observable anymore. Even though the Book-Shop AIP was good, the full unreliability of the `email` channel transforms it into a bad one. RIVERtools detects the presence of critical points and informs the developer. In Fig. 2(b) the developer updates the MAS partition for decentralized runtime verification purposes, in order to preserve the connectedness for sequence and



(a) (b)

Fig. 2. (a) RIVERtools shows the critical points caused by the absence of the `email` channel; (b) partition updated to allow partially decentralized runtime verification.

unique point of choice at least at the "agent subset" scale, given they cannot be preserved at the individual agent scale. Not all the agents subsets (to be more precise, the agents partitions) are admissible: in this scenario a correct partition is for example `[[alice, dave] [barbara, emily] [carol, frank]]`, which is one of those computed and output by RIVERtools.

4 Conclusions

RIVERtools supports the development of AIPs modeled as trace expressions via a user-friendly GUI. It implements static controls to check if a protocol is ugly, bad or good. Since bad protocols reflect realistic situations, where the properties which ensure protocol enactment are violated because of communication channel unreliability, we claim that it is worth keeping them and studying under which conditions they can be used to (partially) decentralize the runtime verification process. RIVERtools supports the MAS developer also in this respect.

References

1. Ancona, D., Briola, D., Ferrando, A., Mascardi, V.: Global protocols as first class entities for self-adaptive agents. In: AAMAS 2015, pp. 1019–1029. ACM (2015)
2. Ancona, D., Ferrando, A., Franceschini, L., Mascardi, V.: Coping with bad agent interaction protocols when monitoring partially observable multiagent systems. In: PAAMS 2018 (2018)

3. Ancona, D., Ferrando, A., Mascardi, V.: Comparing trace expressions and linear temporal logic for runtime verification. In: TPFM 2016. Springer (2016)
4. Ancona, D., Ferrando, A., Mascardi, V.: Parametric runtime verification of multiagent systems. In: AAMAS 2017, pp. 1457–1459. ACM (2017)
5. Ferrando, A.: RIVERtools: an IDE for RuntIme VERification of MASs, and beyond. In: PRIMA Demo Track 2017 (2017)
6. Ferrando, A., Ancona, D., Mascardi, V.: Decentralizing MAS monitoring with DecAMon. In: AAMAS 2017, pp. 239–248. ACM (2017)

Developing Agent-Based Pervasive Mixed Reality Systems: The MiRAgE Framework

Angelo Croatti(✉) and Alessandro Ricci

Computer Science and Engineering Department (DISI),
Alma Mater Studiorum – University of Bologna, via Sacchi, 3, Cesena, Italy
{a.croatti,a.ricci}@unibo.it

Abstract. The impressive development of smart technologies is reducing the gulf between the physical and the digital matter, between what is real and what is virtual. Recent technological advances in the Mixed Reality (MR) context – both tools, such smart-glasses/visor, and software frameworks/libraries – are bringing new opportunities for research in agents and Multi-Agents Systems (MAS), in particular to explore their application in the design and development of mixed reality based smart intelligent environments that we can call *augmented worlds* (AWs). The goal of this demo is to present a first prototype of a framework called MiRAgE to build agent-based pervasive mixed reality software systems, incrementally showing its potentialities and proposed abstractions for the design of such class of systems.

Keywords: Augmented/Mixed Reality · Agents
Multi-Agent Systems · Augmented worlds

1 Introduction

In recent years, augmentation technologies witnessed an impressive development and progress, making them ready for being used out of research labs. With the term "augmentation technologies" here we mean those ICT technologies augmenting the functionalities/services of the physical environment (i.e., pervasive computing [9]), of human capabilities (i.e., wearable computing [8]) and of the physical reality (i.e., Augmented and Mixed Reality [1]) by means of a computational layer. In particular, state-of-the-art of Augmented Reality (AR) and Mixed Reality (MR) technologies – such as, e.g., Microsoft Hololens[1] and Meta2[2] – allow for mixing physical environments with fairly sophisticated holograms that human users can perceive and interact with by means of wearable visors. These technologies can have a huge impact from an application point view, enhancing human cognitive capabilities and allowing for rethinking the way in which people work, interact and collaborate, starting from the industrial sector as envisioned in the idea of Industry 4.0 [7].

1 www.microsoft.com/hololens.
2 www.metavision.com.

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 301–304, 2018.
https://doi.org/10.1007/978-3-319-94580-4_25

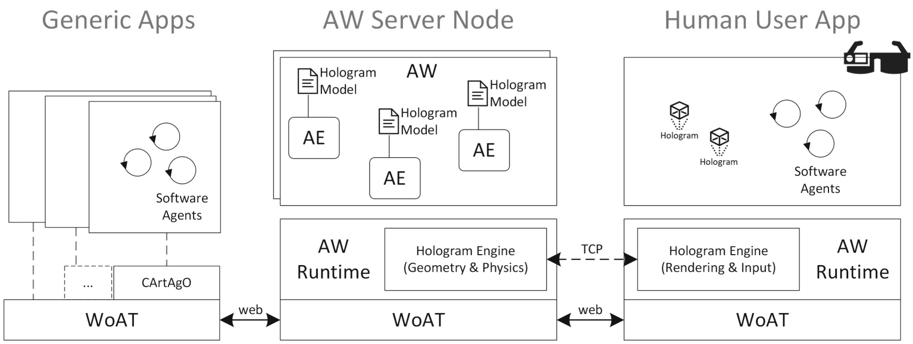


Fig. 1. Logical architecture of the MiRAgE framework.

The Augmented World conceptual framework [5,6] aims at exploring the use of the agent paradigm to design and develop agent-based pervasive mixed reality systems – called *augmented worlds* (AWs) – integrating both concepts and technology from Augmented/Mixed Reality and from Pervasive Computing/IoT. To explore in practice the design and development of AW, possibly in different application domains, a technological framework called MiRAgE has been devised. Key requirements for the platform were: to support heterogeneous agent technologies to implement the agents inside an AW; to reuse and exploit state-of-the-art AR/MR enabling technologies, properly encapsulated so as to keep the agent-oriented level of abstraction defined by the AW conceptual framework.

In this demo we show a first prototype of the MiRAgE framework at work, providing an example of AW programming using well-known cognitive (BDI) agent technologies.

2 Main Purpose

MiRAgE – Fig. 1 shows an abstract architecture – provides an API and a runtime to develop and run Augmented Worlds. According to the Augmented World conceptual framework [5], an agent-based pervasive mixed reality system (i.e., an AW) is modelled as a multi-agent system – possibly open, heterogeneous, distributed – situated into an application/virtual environment mapped onto some physical space. Such environment is composed by *augmented entities* (AEs), as computational objects located in some specific point of the physical space – providing some action interface to agents – and possibly featuring an *hologram* as a AR/MR representation to be perceived by human users situated in the same environment. An AE can be used to model also real objects belonging the physical environment, functioning as a *mirror* or avatar inside the AW. The framework provide an API to implement AEs in a similar way to objects in mainstream object-oriented programming (OOP) languages.

Agents can dynamically join an AW from any Internet node[3] exploiting the Web as underlying protocol—this layer is called Web of Augmented Things (WoAT), being an extension of the Web of Thing (WoT) idea [4]. So heterogeneous agent platforms and technologies can be used to develop agents working inside an AW.

Multiple human users can enter and work inside as AW—along with e.g. their personal assistant agents running on their mobile/wearable devices. Each human user is modelled by the framework as a first-class entity [5], so that an augmented entity associated to the user is automatically created, making it possible to track her position and actions.

As shown in Fig. 1, a MiRAgE node is composed by (1) a AW Runtime, where both AEs are in execution and where agents can interact with, and (2) an Hologram Engine to offer mixed reality experience, defining and shaping holograms for AEs and proposing them to a multi-user environment, interacting in the AW Runtime context. The Hologram Engine encapsulates and hides the AR/MR enabling technologies, in particular in current version of MiRAgE Unity3D[4] and Vuforia[5] are exploited.

3 Demonstration

The objective of the demo is showing and experiencing some main aspects of Augmented Worlds and Augmented Worlds programming exploiting the API and runtime provided by the MiRAgE framework.

We start from an empty AW coupled to some physical environment and then we incrementally enrich it by coding and spawning on-the-fly some cognitive agents that join the AW and create, share, move, interact with augmented entities – either predefined or defined on the fly to the purpose – as bricks of the application environment where they are situated. Augmented entities will range from simple state-less objects such as a message to state-full objects providing an interface – with actions and an observable state – to be eventually exploited by the agents inside the AW. Some of the augmented entities are coupled to entities of the physical world, mirroring them in the AW and making the observable/actionable from the agents inside the AW—producing effects on the physical world.

Multiple human users could enter dynamically the AW so as to perceive and interact with holograms associated to the augmented entities, by means of proper AR/MR devices (from simple smartphones to AR headsets). Agents inside the AW will be capable of perceiving and observing human users' actions, position and behaviour inside the AW, being them coupled to specific augmented entities part of the environment.

A main objective of the demo is to show the effectiveness of the MiRAgE API for implementing the cognitive/BDI agents. Being based on Web API –

[3] Properly ruled by security and access control policies.
[4] http://unity3d.com.
[5] www.vuforia.com.

following the WoAT idea [4] – the MiRAgE platform is independent from the specific platform/technology used to develop the cognitive agents—in the demo, the JaCaMo [2] platform based on Jason/AgentSpeak(L) BDI agents is used. Yet the AW meta-model and the corresponding API has been developed with cognitive/BDI agents in mind.

4 Conclusions

The Augmented World approach aims at enriching existing literature about agent-based Augmented/Mixed Reality Systems (e.g., [3]). The MiRAgE framework provides an API and a runtime to develop and run augmented worlds, providing high level abstractions to investigate and integrate agents and MAS in pervasive mixed reality complex systems.

The first prototype of the framework[6] requires improvements, especially to make it widely available to be considered as a reference platform for agent community to develop mixed reality based systems. Despite this, the current state of the prototype provides a first taste of opportunities that can be obtained for agents and MAS and functionalities that can be easily reached and integrated.

References

1. Azuma, R.T.: A survey of augmented reality. Presence Teleoperators Virtual Environ. **6**(4), 355–385 (1997)
2. Boissier, O., Bordini, R.H., Hübner, J.F., Ricci, A., Santi, A.: Multi-agent oriented programming with JaCaMo. Sci. Comput. Program. **78**(6), 747–761 (2013)
3. Campbell, A.G., Stafford, J.W., Holz, T., O'hare, G.M.: Why, when and how to use augmented reality agents (AuRAs). Virtual Real. **18**(2), 139–159 (2014)
4. Croatti, A., Ricci, A.: Mashing up the physical and augmented reality: the Web of Augmented Things Idea. In: Eighth International Workshop on the Web of Things, WoT 2017, 22 October 2017, Linz, Austria. ACM, New York (2018). (Under publication)
5. Croatti, A., Ricci, A.: A model and platform for building agent-based pervasive mixed reality systems. In: Proceedings of the 16th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2018) (2018)
6. Ricci, A., Piunti, M., Tummolini, L., Castelfranchi, C.: The mirror world: preparing for mixed-reality living. IEEE Pervasive Comput. **14**(2), 60–63 (2015)
7. Schwab, K.: The Fourth Industrial Revolution Hardcover. Crown Business, New York (2017)
8. Starner, T.: Project glass: an extension of the self. IEEE Pervasive Comput. **12**(2), 14–16 (2013)
9. Weiser, M.: The computer for the 21st century. SIGMOBILE Mob. Comput. Commun. Rev. **3**, 3–11 (1999)

[6] The current state of the MiRAgE framework can be found at the following BitBucket link: https://bitbucket.org/account/user/awuniboteam/projects/AW.

Generating Unemployment Expectations of the "Man in the Street"

Luca Gerotto(✉) and Paolo Pellizzari(✉)

Department of Economics, Ca' Foscari University,
Cannaregio 873, 30121 Venice, Italy
{luca.gerotto,paolop}@unive.it

Abstract. Why are unemployment expectations of the "man in the street" markedly different from professional forecasts? We present an agent-based model to explain this disconnection using boundedly rational agents with different levels of education.

Keywords: Agent-based modeling · Bounded rationality
Unemployment expectations

1 Introduction

Why are unemployment expectations widely heterogeneous among the "men in the street" and strikingly different from the ones produced by highly regarded institutions? The question is extremely important, as economic decisions of households strongly depend on their expectations.

2 Main Purpose

Empirical surveys show that economic agents produce vastly different sets of unemployment forecasts that appear to be quite disconnected from professional expectations. Even if the assumption of perfect rationality of agents is still present in the majority of scholarly work, this is at odds with the observed data.

The present model assumes that households are boundedly rational in the sense that their expectations are often based on wrong grounds, due to their limited ability or willingness to detect (and digest!) relevant information. The framework is related to the epidemiological metaphor used in Carroll [1]: information is modeled like a virus, which may "infect" the agents. Households can probabilistically absorb the information at time t by a common source (say, mass media coverage) and react accordingly, or keep using the obsolete information or even act on the basis of spurious facts. We will explore the implications of different degrees of (bounded) rationality on aggregate expectations.

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 305–308, 2018.
https://doi.org/10.1007/978-3-319-94580-4_26

3 Demonstration

Assume there are N agents, whose education level is $edu_i \in E = \{lths, hs, col\}$, with the strings denoting *Less than High School, High School* and *College*, respectively. Agents develop their expectations differently as a function of their education level.

At each time step agents can either get an informative signal o_t, with some probability $\lambda_i = \lambda(edu_i)$ that depends on their education or remember the old signal $s_{i,t-1}$ if they get no fresh news, or forget even the old signal. Formally the signal is given by:

$$s_{it} = \begin{cases} o_t & \text{with probability } \lambda_i \\ s_{i,t-1} & \text{with probability } \beta(1-\lambda_i) \\ 0 & \text{with probability } (1-\beta)(1-\lambda_i) \end{cases}, \tag{1}$$

where o_t is the latest professional forecast released by the media and β is the probability to remember the past signal if one does not obtain a fresh forecast.

Once the (informative or uninformative) signal is available, agents convert it into an answer ans_{it} to the question "How do you expect the number of people unemployed to change over the next 12 months?". Similarly to the Michigan Survey or the European Commission's Consumer Survey, answers are encoded with integers in $\{-2, -1, 0, +1, +2\}$ corresponding to qualitative answers {decrease sharply, decrease slightly, remain the same, increase slightly, increase sharply}, respectively.

We assume that agents stubbornly report $ans_{it} = 2$ with probability $\mu_i = \mu(edu_i)$; otherwise they "translate" their signal based on its perceived magnitude in the following way:

$$ans_{it} = \begin{cases} +2 & \text{with probability } \mu_i \\ f(s_{it}|\gamma) & \text{with probability } 1-\mu_i \end{cases}. \tag{2}$$

The translation function f depends on a threshold $\gamma > 0$ that shapes 5 ranges of values driving the interpretation of the signal: the details are given in the companion paper "Unemployment expectations in an agent-based model with education" but, in essence, if the signal is large and positive, the agent will claim that unemployment will increase sharply; if the signal is intermediate and positive, the milder conclusion is that unemployment is about to increase slightly; finally, if the signal is small, no change is reported (and a similar mechanism for negative signals will induce decrements of varying sizes).

At the end of period t, when all answers $ans_{it}, i = 1, \ldots, N$, are available, it is straightforward to compute an aggregate balance index:

$$x_t = \frac{100}{N} \sum_{i=1}^{N} ans_{it}.$$

We start from the assumption that the more educated are more "rational", in the sense that they read newspapers more frequently and they have higher attention (i.e., higher $\lambda(edu)$) and higher trust (i.e., lower $\mu(edu)$) towards the forecasts released by the mass media. The values of the parameters are reported in Table 1.

Table 1. Calibrated values of μ and λ

Parameter	$\lambda(lths)$	$\mu(lths)$	$\lambda(hs)$	$\mu(hs)$	$\lambda(col)$	$\mu(col)$
Value	0.1	0.3	0.2	0.2	0.3	0.1

We present five simulations. The baseline scenario is simulation (i), where $\beta = 0.8$, $\gamma = 0.4$ and, of the N agents, one third is $lths$, one third hs and one third is col. Then, in simulation (ii) two thirds of the population are $lths$ and one third is hs, while in (iii) one third is hs and two thirds are col. Then, in simulation (iv) $\beta = 0.6$ and in simulation (v) $\beta = 1$.

We present a simulation with a total of 3000 agents for 200 periods. o_0 is drawn from an uniform $U[-0.5, 0.5]$, then $o_t = 0.8o_{t-1} + \epsilon_t$, where $\epsilon_t \sim N(0, 0.25)$.

Table 2. Calibration and summary statistics

	β	% $lths$	% hs	% col	Min	1st Qu	Median	Mean	3rd Qu	Max	St. Dev.	$\rho(x_t, x_{t-1})$
i	0.8	33.3	33.3	33.3	−46.00	−4.56	37.17	38.44	81.75	130.70	51.27	0.953
ii	0.8	66.7	33.3	0	−9.93	23.65	50.95	52.39	82.87	119.60	35.94	0.960
iii	0.8	0	33.3	66.7	−89.13	−31.14	23.33	24.49	80.39	143.90	66.85	0.947
iv	0.6	33.3	33.3	33.3	−27.13	7.37	39.08	38.79	69.76	104.60	38.87	0.927
v	1	33.3	33.3	33.3	−94.93	−26.96	39.82	38.15	94.27	193.20	77.83	0.975

Table 2 contrasts the distributions under the five simulations. Comparing the baseline with simulation (ii) [(iii)], a decrease [increase] in the proportion of educated individuals leads to an aggregate expectation which is more [less] pessimistic (higher [lower] mean), less [more] dispersed (lower [higher] standard deviation) and slightly more [less] persistent (higher [lower] one-lag autocorrelation $\rho(x_t, x_{t-1})$).

Then, comparing the baseline with simulation (iv) [(v)], a decrease [increase] in the probability of remembering the past information has no appreciable effect on the mean, but leads to a lower [higher] dispersion (lower [higher] standard deviation) and lower [higher] persistence (lower [higher] autocorrelation).

Figure 1 plots the results of the five simulations: in the left panel simulations (i), (ii) and (iii) are compared, (i), (iv) and (v) are contrasted in the right panel. Dotted lines delimit the range of values attained by the different simulations. Note also that these ranges are much narrower than the theoretical span of $[-200, 200]$ but, actually, much larger than the ranges observed in the real data analyzed in the companion paper.

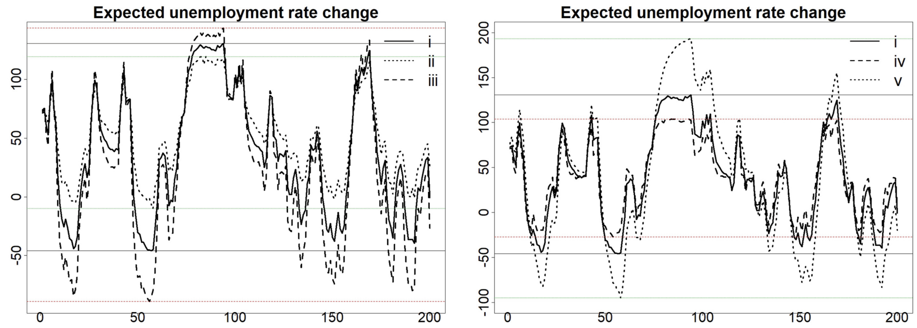


Fig. 1. Households unemployment expectations balance

The practical session will also contain a demonstration of the use of NetLogo's [2] `BehaviorSpace` to calibrate the parameters to empirical data. We will show how to modify the code to relax the assumption that information can *only* be obtained by a central source, allowing for more peer-to-peer interaction. The time series obtained when agents can (copy and) use the information of other households will be discussed and compared with the previous results.

4 Conclusion

The present demonstration, based on (simulated) fictitious series of inputs, has shown ways to reproduce real time series by suitably calibrating the parameters of an agent-based model populated with individuals who have different education and may, to different extents, fail to update information or get it from other agents. The aggregate expectations depend on the degree of (bounded) rationality of the agents and are affected in particular by the presence of an average, or "permanent", level of pessimism that prompts agents to declare that unemployment will increase sharply disregarding the professional forecasts (whatever is their content).

References

1. Carroll, C.D.: The epidemiology of macroeconomic expectations. In: Blume, L.E., Durlauf, S.N. (eds.) The Economy as an Evolving Complex System, III: Current Perspectives and Future Directions, pp. 5–29. Oxford University Press, Oxford, NewYork (2006)
2. Wilensky, U.: NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston (1999). http://ccl.northwestern.edu/netlogo/

MASAP: Multi-Agent Simulation of Air Pollution

Sabri Ghazi[1(✉)], Julie Dugdale[2], and Tarek Khadir[1]

[1] Laboratoire de Gestion Electronique de Documents, Department of Computer Science, University Badji Mokhtar, PO-Box 12, 23000 Annaba, Algeria
{Ghazi,Khadir}@labged.net

[2] LIG, University Grenoble Alpes, Grenoble, France
Julie.Dugdale@imag.fr

Abstract. This paper presents MASAP, a Multi-Agent Simulator of air pollution. The aim is to simulate the concentration of air pollutants emitted from sources (e.g. factories) and to assess the efficiency of air quality controlling policies. The pollutions sources are modelled as agents. Autonomously, the agents try to achieve their goals (increasing production, which has the side effect of increasing pollution) and cooperate with others agents by altering their emission rate according to the pollution level. The rewards/penalties are influenced by the pollutant concentration which is, in turn, determined using climatic parameters. In order to give predictions about the concentration of pollutants: Particulates Matter (PM10), Sulphur Oxide and Dioxide (SO_x), Nitrogen oxides (NO_x) and Ozone: (O_3), a combination between a GPD (Gaussian Plume Dispersion) algorithm and an ANN (Artificial Neural Network) is used. The prediction is calculated using real data about climatic parameters (wind speed, humidity, temperature and rainfall). Every agent cooperates with its neighbours that emit the same pollutant, and it learns how to adapt its strategy to earn reward and avoid penalties. When the pollution reaches a peak, agents are penalised according to their participation. The Simulator helps the decision makers to assess their air pollution controlling policies and oversee the results of their decision.

Keywords: Multi-agent system · Multi-agent based simulation
Agent-based modelling · Environmental modelling · Air pollution
Air quality · PM10

1 Introduction

Many cities in the world are facing the deterioration of air quality. Air pollution has a direct influence on health [1]. The increase of pollutant concentration in the air should be deeply studied before the establishment or the expansion of urban or industrial activities. Air pollution simulation and decision support tools can help decision-makers to put policies for environmental management and to predict the impact of their decisions. To study air pollution, several modelling approaches have been proposed.

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 309–313, 2018.
https://doi.org/10.1007/978-3-319-94580-4_27

Most of them, [2–4] to cite a few, are mainly focused on the physical and chemical aspects of air pollution. These models do not take in consideration human-decision factors. Air pollution is by nature distributed and includes the interaction of individuals involved in the exploitation of a dynamic ecological resource which is the air. Road traffic, industrial and agricultural activities are considered as the major sources of air pollution. All these activities are controlled by humans; therefore, including the human decision factors in the modelling of air pollution is important.

MAS (Multi-Agent Simulation) is an appropriate method for modelling systems that includes human actors and environmental issues [5]. They allow us to model the decision of human actors sharing the use of environmental resources. [6] presents a review of recent MAS models used to investigate the socio-environmental problems.

2 The System Architecture

Our simulation loop can be schematised as shown in Fig. 1. Agent decides to increase or to decrease the emission rate of the sources. Then the dispersion algorithm is used to compute the dispersion. The pollutant dispersion and climatic parameters are used to forecast the *k* hours ahead air pollution concentration and air quality. According to these forecasts, agents are rewarded or penalised. Agents then adapt their strategies to earn more reward and avoid penalties.

MASAP is composed by four main components: The Multi Agent System; The GDP-ANN Module; The core model which contains data loading and pre-processing routines and finally the user interface as shown in Fig. 2.

The Multi-agent system is a network of *N* agents representing the *N* emission sources, for each one is represented by: the emitted pollutant, the emission rate, the earned reward and its position. The user can choose between four cooperation strategies according to it, agents will interact: Centralized, EG Evolutionary Game with cumulative penalties, Evolutionary Game with no cumulative penalties.

GDP-ANN module is the combination between a Gaussian Plume Dispersion algorithm as is presented in Eq. (1) and an Artificial Neural Network. The ANN model receives the dispersion value and the climatic condition as input in order to gives prediction about the pollutant concentration.

$$C(x,y,z,H) = \frac{er_{i,t} * D}{2\pi U_t \, \sigma_y \sigma_z} * e^{-\frac{y^2}{2*\sigma_y^2}} * \left[(e^{-(\frac{(z-H)^2}{2\sigma_z^2})}) + (e^{-(\frac{(z+H)^2}{2\sigma_z^2})}) \right] \tag{1}$$

This means that, the concentration of the pollutant at point (x, y, z) is calculated according to: $er_{i,t}$: the emission rate in kilograms per hour of the source i in time step t. U_i: the wind speed in metres per second at time step t, $\sigma_y \sigma_z$: the standard deviation of the concentration distributions in crosswind in vertical direction.

3 Example of Scenario About Annaba City (North-East of Algeria)

In this section we present a scenario based on real data about the city of Annaba. Which is known to be one of the most industrialized Algerian city. The used data set is about the period of 2 year from 2003 to 2004. Is contains hourly measures about four pollutants PM10, NO_x, SO_x and Ozone.

The scenario is resumed in Table 1, it includes 300 agents (emission sources), with the goal to reach an air quality of index Good, according to Algerian air pollution standards. Some results about air quality evolution during the time of simulation are presented in Fig. 3. As shown, the system is able to show how environmental policies can affect the evolution of air quality. The system could be used for training purpose and as decision support tool. The simulator is designed in generic way and it could be adapted for other city and other type of pollution such as water pollution. The Simulator is designed using JADE framework and it is distributed under the GPL (General Public Licence).

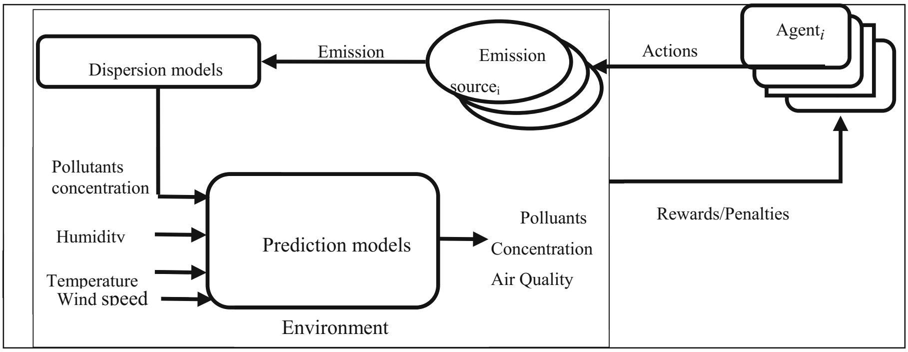


Fig. 1. MASAP architecture.

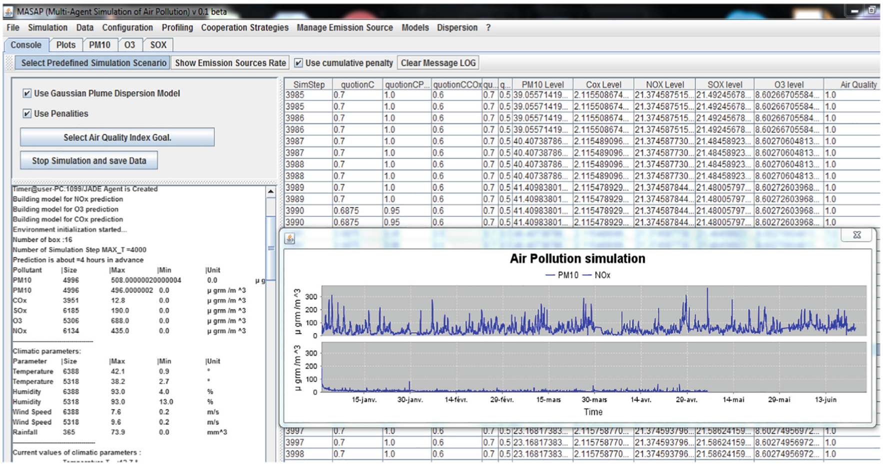


Fig. 2. User interface of MASAP.

Table 1. Parameters values for the simulation scenario.

Parameter name	Value
Number of PM10, SO_x, NO_x sources	100 for each pollutant
Max emission rate	2000 (g/h).
Goal PM10 level	20 $\mu g/m^3$
Goal SO_x level	30 $\mu g/m^3$
Goal NO_x level	45 $\mu g/m^3$
Goal O_3 level	45 $\mu g/m^3$
Number of memory steps (*L*)	4 steps
Initial proportion of cooperating agents	0.5
Air quality at t = 0	2 (Good)
Total simulation time	4900 h
Prediction horizon	2 h in advance

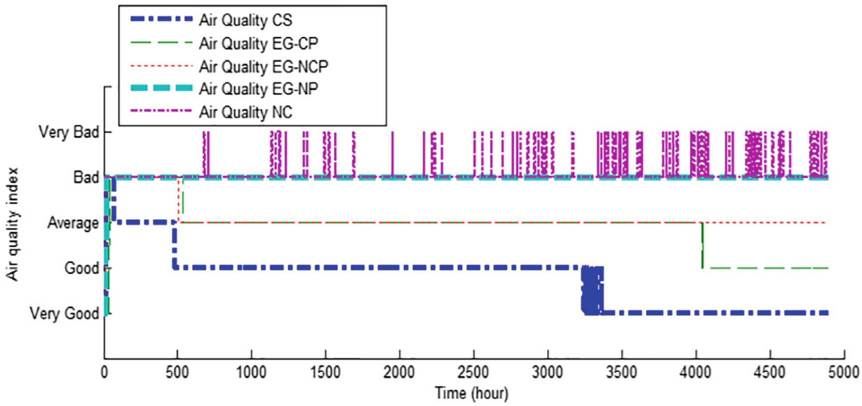


Fig. 3. Air quality of each cooperation strategies compared with NC (no-cooperation).

References

1. WHO (World Health Organisation): Ecosystems and Human Well-Being: Health Synthesis. WHO Library Cataloguing-in-Publication Data (2005)
2. Kryza, M., Netzel, P., Drzeniecka-Osiadacz, A., Werner, M., Dore, A.J.: Forecasting the ozone concentrations with WRF and artificial neural network based system. In: Steyn, D., Builtjes, P., Timmermans, R. (eds.) Air Pollution Modeling and its Application XXII, pp. 605–609. Springer, Dordrecht (2014). https://doi.org/10.1007/978-94-007-5577-2_102
3. Holmes, N.S., Morawska, L.: A review of dispersion modelling and its application to the dispersion of particles: an overview of different dispersion models available. Atmos. Environ. **40**(30), 5902–5928 (2006)
4. Lushi, E., Stockie, J.M.: An inverse Gaussian plume approach for estimating atmospheric pollutant emissions from multiple point sources. Atmos. Environ. **44**(8), 1097–1107 (2010)

5. Aulinas, M., Turon, C., Sànchez-Marrè, M.: Agents as a decision support tool in environmental processes: the state of the art. In: Cortés, U., Poch, M. (eds.) Advanced Agent-Based Environmental Management Systems. Whitestein Series in Software Agent Technologies and Autonomic Computing, pp. 5–35. Birkhäuser, Basel (2009). https://doi.org/10.1007/978-3-7643-8900-0_2
6. Ghazi, S., Khadir, T., Dugdale, J.: Multi-agent based simulation of environmental pollution issues: a review. In: Corchado, J.M., et al. (eds.) PAAMS 2014. CCIS, vol. 430, pp. 13–21. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07767-3_2

Check for updates

Analysis of Agent-Based Parallelism for Use in Clustering and Classification Applications

Collin Gordon(✉) and Munehiro Fukuda

University of Washington, Bothell, WA 98011, USA
{colntrev,mfukuda}@uw.edu
http://depts.washington.edu/dslab/MASS.html

1 Introduction

The Multi-Agent Spatial Simulation (MASS) library, which we have developed at University of Washington Bothell, provides the efficiency of agent-based parallelization while abstracting the parallel environment for the agent-based model [1]. It provides a middle layer that only requires the user to contribute code relevant to their project without having to manage the details of parallelization.

MASS is made up of two different component objects: *Agents* and *Places*. *Agents* are serializable, migratable objects with two states: static and dynamic. *Places*, on the other hand, are objects that behave statically in a grid formation. Each *place* can facilitate communication with its immediate neighbors and any *Agents* that currently reside on it.

2 Purpose

The purpose of this demonstration is to prove the performance ability of agent-based parallelism when applied to two machine learning problems: K Means Clustering and K Nearest Neighbor classification.

Exploration of these algorithms with MASS has incorporated the research of biological optimization heuristics due to the inherent use of agents as members of a swarm or pack. The algorithms primarily explored were the Grasshopper Optimization and Particle Swarm Optimization algorithms. These algorithms were used to develop the Best Tool Clustering Algorithm for the MASS library [2,3].

The Best Tool algorithm performs many instances of K Means clustering over a data set. Agents behave statically and share the computational load with Places. Places, in this algorithm, read data and pass the data to the Agents who subsequently perform clustering. Figure 1 represents a map of the program with three methods highlighted: *kmeans*, *newCentroid*, and *collectBest* which perform K Means clustering, calculate new centroids, and return the results to the main program respectively.

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 314–317, 2018.
https://doi.org/10.1007/978-3-319-94580-4_28

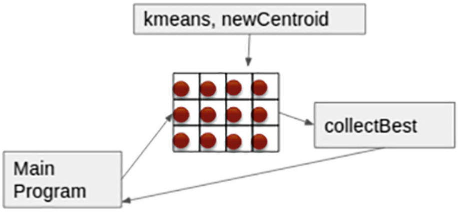


Fig. 1. The Best Tool algorithm

The K Nearest Neighbor algorithm uses a heuristic that we developed in the pursuit of a suitable clustering implementation. The heuristic mimics the behavior of ripples on the water. As agents send a ripple out, places containing data points will send their information back to the agent. Each agent that issues a ripple will dynamically spawn new agents which mark the places used for classification. Figure 2 demonstrates this algorithm.

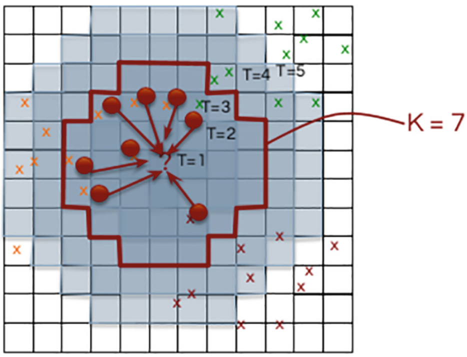


Fig. 2. The agent-based K Nearest Neighbor algorithm

3 Demonstration

The following are results from demonstrations performed on the University of Washington Bothell's general purpose linux cluster. The linux cluster contains 16 Dell Optiplex 710 desktops, each with an Intel i7-3770 Quad-Core CPU at 3.40 GHz and 16 GB RAM.

3.1 Best Tool

The Best Tool demonstration generated a large number of three centroid representations using a randomly generated set of 500 cartesian points. Finding an ideal representation of the data was not taken into account because it has been proven that similar optimizations possess a high probability of producing an optimal representation of the data set [2,3]. Therefore, the focus was whether MASS can significantly improve the number of centroid combinations generated and the efficiency of their calculation.

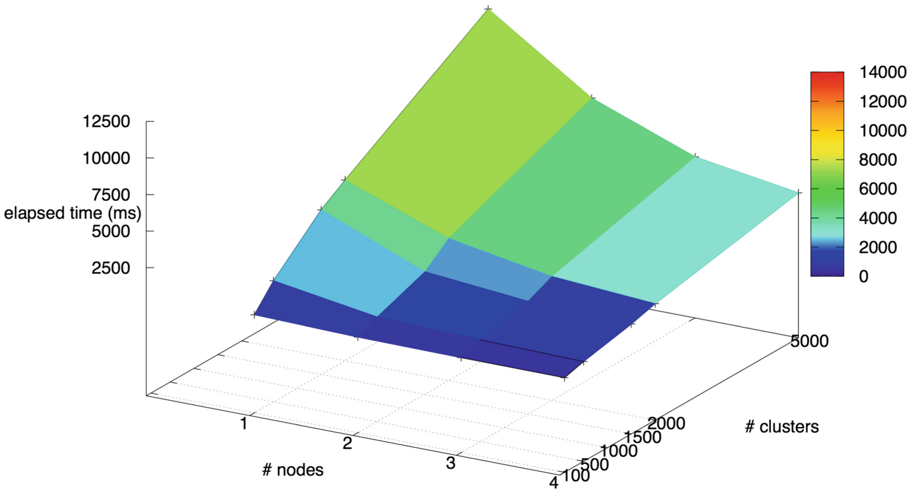


Fig. 3. Graph showing the performance results of the sequential, 2, 3, and 4 node executions of the Best Tool Algorithm.

As can be seen in Fig. 3, MASS Best Tool can perform up to 3.3 times faster than an equivalent sequential program. Additionally, the amount of representations computed is significantly higher than either GOA or PSO. In the demonstration, MASS computed upwards of 15,000 different representations for the data set. Similar algorithms computed a maximum of 30 such representations [2,3].

3.2 KNN

Figure 4 shows a graph comparing execution time for an increasing amount of data between a sequential program and MASS on a single node. This graph is intended to demonstrate the effects of the agent spawning overhead on the performance of the algorithm.

For MASS agents to spawn children, they must initialize brand new agents with the same properties as the parent. Depending on the algorithm, this could mean making a copy of an agent that takes up a lot of room in memory. This combined with any additional agent logic significantly increases the computational complexity of any of the dynamic agent algorithms using MASS.

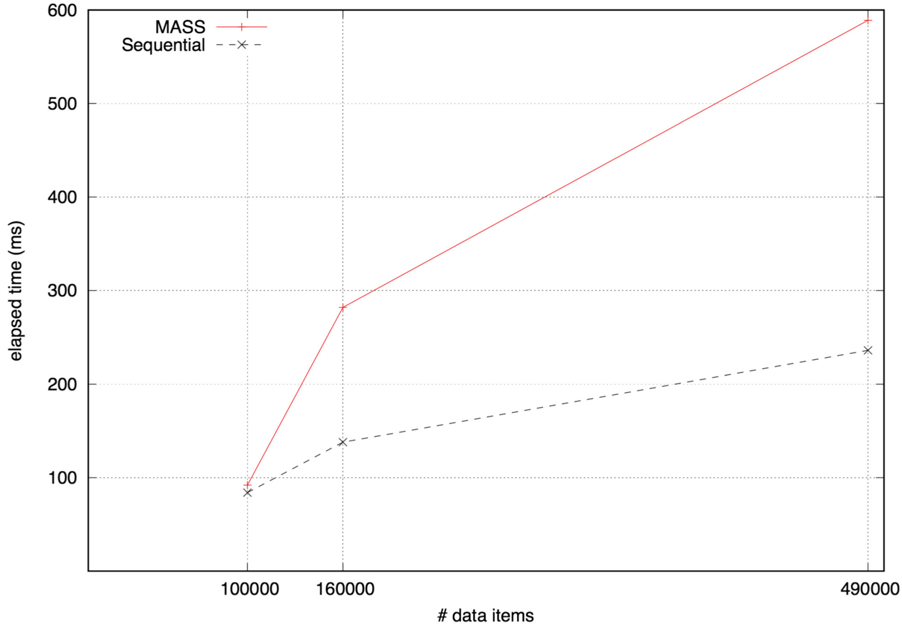


Fig. 4. Graph showing KNN performance

4 Conclusion

These demonstrations prove that not only is parallelization of clustering and classification algorithms possible with MASS, but that it can result in a significant performance improvement. Future work will be done with the Best Tool algorithm to test it using real, labeled data sets to see if not only efficiency is improved, but also the ability to find an optimal representation for the given data. The final test of the Best Tool algorithm will use unlabeled data to see if an accurate representation can be achieved and used successfully in KNN classification. The KNN algorithm will be used as a benchmark for research into the reduction of agent overhead in the MASS library. Finally, once the validity of both algorithms are sufficiently established, work to explore the possibilities of using agents for other machine learning algorithms will be performed.

References

1. Fukuda, M.: MASS: a parallelizing library for multi-agent spatial simulation. http://depts.washington.edu/dslab/MASS
2. Lukasik, S., Kowalski, P.A., Charytanowicz, M., Kulczycki, P.: Data Clustering with Grasshopper Optimization Algorithm
3. Van der Merwe, D.W., Engelbrecht, A.P.: Data clustering using particle swarm optimization. In: The 2003 Congress on Evolutionary Computation, CEC 2003, vol. 1, pp. 215–220. IEEE (2003)

HVUAN – A Rapid-Development Framework for Spanish-Speaking Virtual Humans

David Herrera(✉), Luis Herrera, and Yerson Velandia

Universidad Antonio Nariño, Bogotá, Colombia
{herrera78,lherrera081,yevelandia}@uan.edu.co

Abstract. We present a rapid-development framework for Spanish-speaking virtual humans (VH). Our approach tries to minimize the effort of working with VH systems while preserving customizability of the interaction. We effortlessly generate a 3D representation of the VH and blendshapes for lipsync and facial expression using MakeHuman and a Blender plugin. We animate the VH using captured motion data using a Kinect and affordable motion capture software. The framework uses Cortana for Windows 10 to handle Spanish speech recognition and Excel and/or yEd to specify the dialog interaction in graph format, where nodes contain agent, audio, and animation information and edges hold grammar options for Cortana. The system can handle recognition while it speaks, allowing for interrupts, and automatically generates gestures or speech when only text is specified. This system should work for any language for which Cortana supplies a speech recognizer.

Keywords: Virtual human system · Rapid development frameworks
Embodied conversational agent · Nonverbal behavior generator
Gesture · Dialogue manager

1 Introduction

Embodied conversational agents (ECA) or Virtual Humans (VH) are autonomous digital characters programmed to interact with users both verbally and non-verbally. Non-verbal interaction refers to behaviors such as gesture, gaze, and expressions that co-occur with speech. ECAs are used for education, entertainment, training and research, and their real-world applications range from virtual [1, 2] to pedagogical agents [3, 4] and military training [5, 6].

These systems are complex, incorporating many disciplines, requiring knowledge of automated speech recognition, natural language understanding, dialog management, nonverbal behavior generation, speech synthesis, character animation, and even computer vision. Traditionally, these characters were cost-prohibitive to develop, let alone effectively and efficiently. Fortunately, releases such as the ICT Virtual Human Toolkit[1] and UTEP's Virtual Agent Framework[2] (VAIF) [7] have reduced this cost, enabling non-expert users to create virtual humans for their needs without too much

[1] https://vhtoolkit.ict.usc.edu/.
[2] https://github.com/isguser/VRAIN.

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 318–321, 2018.
https://doi.org/10.1007/978-3-319-94580-4_29

effort, exploring different types of virtual human systems, such as virtual listeners, question-answering characters, and virtual role-players.

However, current versions have been designed for rapid-development of question-answering characters using American English as its language. Users of other languages still have to overcome some barriers such as integrating an automated speech recognizer for their language, and creating nonverbal behaviors for the ECA.

In this paper, we present an ongoing effort to adapt VAIF for use with Colombian Spanish. Non-expert users can generate a 3D representation of the VH using Make-Human[3] and assign blendshapes for lipsync and facial expression with a Blender[4] plugin. Gesture creation for the VH is accomplished with motion data capture with a Kinect and affordable markerless motion capture software. In Excel and/or yEd[5], users specify the dialog interaction in graph format including agent, audio, and animation information in nodes as well as grammar options for Cortana in edges. The system can handle recognition while it speaks, allowing for interrupts, and automatically generate gestures or speech when only text is specified with a word-to-gesture mapper in xml format. Although the framework uses Cortana for Windows 10 to handle Spanish speech recognition, it can seamlessly be adapted to any language which Cortana supports.

1.1 Preparing the Virtual Human

Creating a new 3D character with its own gestures usually requires a graphics designer. Humano Virtual Universidad Antonio Nariño (HVUAN) sidesteps this need by using MakeHuman to generate a VH. Using the MHX2[6] file exchange format, we can import our VH into Blender to further enhance it with visemes for lipsync and facial expressions using the MHX2 plugin. We can finally export our VH to Unity using fbx format to use with VAIF.

1.2 Preparing the Gestures

HVUAN sidesteps the need for an animator by using iPiSoft[7], an affordable markerless motion capture software. It is slightly ungainly for generating hand and wrist gestures, so we are currently looking at Brekel[8] as an alternative, along with various open-source options. Using a Kinect, users can simultaneously capture gesture and audio data, or alternatively, capture gesture data in fbx format to be assigned to lexical affiliates in xml format to feed the gesture generator.

---

3 http://www.makehuman.org/.

4 https://www.blender.org/.

5 https://www.yworks.com/products/yed.

6 https://thomasmakehuman.wordpress.com/2014/10/09/mhx2/.

7 http://ipisoft.com/.

8 http://brekel.com/.

2 Main Purpose

The main purpose of HVUAN is to allow the Spanish-speaking research community to quickly and easily create virtual humans that recognize Spanish and produce gestures appropriate to their culture.

Our system builds off of VAIF, a framework that allows the creation of VHs in Unity, specifying dialogs and animations for an agent on a timeline, allowing jumps in the timeline based on speech recognition. However, VAIF works only for American English because of a Unity limitation. To switch to a Spanish recognizer, we elected to use sockets to allow communication between Cortana and Unity. We also elected to specify our dialog using graphs instead of VAIF's timeline, mainly so that we could have more freedom to experiment with dialogue managers. The VAIF timeline is similar to a screenplay, where the agent controls the interaction, limiting the user to respond to VH questions. We wanted to allow the user to jump around the conversation by selecting a topic, so instead we specify states and transitions in Excel using node and edge list format and view the resulting flowchart in yEd. This allows us to design the interaction like a timeline, but allowing conversational jumps by selecting edges based on keywords.

Each node specifies an agent along with a speech file to play or corresponding text to synthesize and an animation file to execute. Omitting the latter generates gestures automatically based on the text and mappings of gestures to words specified in an xml file. Each edge specifies the responses or keywords to activate it and consequently the next state or node it points to. We reduce the workload by enhancing frequent responses such as *yes* and *no* to a list of alternatives, such as *perhaps, maybe, of course, could be* and *never, impossible, forget it* respectively.

Finally, HVUAN has the option of listening while speaking, allowing for a user to interrupt the VH midsentence. This is useful for push-to-talk or headset mic systems.

3 Demonstration

We will demonstrate how users can quickly and easily create a dialogue interaction for question-answering tasks such as product and brand promotions, general service and tour guide information, frequently asked questions, self-service orders, and other easily automated interactions. Users will be able to interact in Spanish with our Virtual Dean of the Faculty of Systems Engineering, a VH that encourages students to consider joining the Faculty and pursue a career in computer science and also promotes the Universidad Antonio Nariño. Users can create their own dialogue interactions with our proposed workflow to substantiate the facility with which these systems can be created.

4 Conclusions

HVUAN is an initial step towards frameworks for rapid-development of foreign language VHs for the international community. We create our VHs with open-source software, generating 3D characters with facial expressions and visemes rather effortlessly. Similarly, we express the dialog interaction with states representing VH discourse and transitions representing human speech input. State or nodes specify speech and animation files, or alternately, word-to-gesture rules can be specified in an xml file for automatic animation, benefitting from increasingly robust markerless motion capture to create culturally-appropriate gesture lexicons for our VH.

Our system relies on Cortana speech recognition for Spanish but can potentially be used for other foreign languages which it supports. Finally, we offer features such as continuous speech recognition for interrupting the VH and user-initiated topics to provide a richer interaction with our virtual humans.

Acknowledgments. We would like to thank Dr. Ivan Gris for assistance with initial versions of HVUAN and Dr. David Novick and UTEP's Advanced Engagement Team (AGENT) for providing their VAIF tool for the latest version of HVUAN. We would also like to thank the Universidad Antonio Nariño in Bogotá, Colombia for funding this project through an internal research grants #2016227.

References

1. Rossen, B., Lok, B.: A crowdsourcing method to develop virtual human conversational agents. Int. J. HCS **70**, 301–319 (2012)
2. Bickmore, T., Bukhari, L., Vardoulakis, L.P., Paasche-Orlow, M., Shanahan, C.: Hospital buddy: a persistent emotional support companion agent for hospital patients. In: Nakano, Y., Neff, M., Paiva, A., Walker, M. (eds.) IVA 2012. LNCS (LNAI), vol. 7502, pp. 492–495. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33197-8_56
3. D'mello, S., Graesser, A.: AutoTutor and affective AutoTutor: learning by talking with cognitively and emotionally intelligent computers that talk back. ACM Trans. Interact. Intell. Syst. **2**(4), 23:1–23:39 (2013). https://doi.org/10.1145/2395123.2395128. ISSN 2160-6455
4. Lane, H.C., Noren, D., Auerbach, D., Birch, M., Swartout, W.: Intelligent tutoring goes to the museum in the big city: a pedagogical agent for informal science education. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) AIED 2011. LNCS (LNAI), vol. 6738, pp. 155–162. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21869-9_22
5. Johnson, W., Valente, A.: Tactical language and culture training systems: using AI to teach foreign languages and cultures. AI Mag. **30**, 72–83 (2009)
6. Campbell, J., Core, M., Artstein, R., Armstrong, L., Hartholt, A., Wilson, C., Georgila, K., Morbini, F., Haynes, E., Gomboc, D., Birch, M., Bobrow, J., Lane, H., Gerten, J., Leuski, A., Traum, D., Trimmer, M., DiNinni, R., Bosack, M., Jones, T., Clark, R., Yates, K.: Developing INOTS to support interpersonal skills practice. In: Proceedings of the Thirty-Second Annual IEEE Aerospace Conference, pp. 1–14 (2011)
7. Gris, I., Novick, D.: Virtual agent interaction framework (VAIF): a tool for rapid development of social agents. Technical report UTEP-CS-18-19, Department of Computer Science, The University of Texas at El Paso (2018)

Check for updates

ATHOS - A Domain-Specific Language for Multi-agent Simulations

Benjamin Hoffmann[1(✉)], Kevin Chalmers[2], Neil Urquhart[2], Thomas Farrenkopf[1], and Michael Guckert[1]

[1] KITE - Kompetenzzentrum für Informationstechnologie, Technische Hochschule Mittelhessen, Gießen, Germany
{benjamin.hoffmann,thomas.farrenkopf,michael.guckert}@mnd.thm.de
[2] School of Computing, Edinburgh Napier University, Edinburgh, Scotland
{k.chalmers,n.urquhart}@napier.ac.uk

Abstract. Creating multi-agent simulations is a challenging task often requiring programming skills at the professional developer level which domain experts scarcely possess. We present a model-driven approach that relieves agent experts from time-consuming, error-prone implementation tasks and allows them to focus on the application itself. With our domain specific language Athos, network-based traffic simulations can be created declaratively. The models are platform independent and executable code can be generated for two popular multi-agent platforms.

Keywords: Domain-specific language · Model-driven development Traffic simulation

1 Introduction

Athos is a DSL that supports the development of MAS simulations. It focuses on traffic scenarios that involve vehicles (agents) with individual behaviour (e.g. finding shortest routes). Athos allows scenario definition in a declarative manner and thus relieves users from complex programming tasks and enables them to focus on *what* to simulate instead of *how* to simulate it (c.f. [1]). Athos simulations are based upon the development of a conceptual model which features aspects from the domain of traffic simulation and optimisation. Relevant elements of this context, e.g. the capacity of certain roads in an area of interest, can be described in an abstract form. Generally, these models do not contain any computational details. From a computational view, they are created on the most abstract level and considered as Computationally Independent Models (CIM) [2].

However, Athos is not fixed on the CIM level. If users need control over computational aspects, they can enrich models with such details for concrete computer simulations. In any case, Athos programs are strictly platform independent models (PIM). The Athos generator processes Athos programs and transforms them into models for a specific simulation platform known as Platform Specific Models (PSM). In order to transform a PIM into a PSM, the generator has to add

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 322–325, 2018.
https://doi.org/10.1007/978-3-319-94580-4_30

platform-specific details to the information drawn from the PIM. This is done by means of code templates which are created for every supported target platform. Currently, Athos features templates for two agent-based simulation platforms. Firstly, the NetLogo[1] platform, whose models follow a procedural paradigm. Secondly, the Repast Simphony[2] platform for which models are constructed in an object-oriented way. As is pointed out by Sansores and Pavón, the creation of models for different target platforms supports validation and verification efforts as both simulation implementations should present equivalent results [4].

The main tool used in the development of the language is the Xtext[3] language workbench (version 2.12). Next to the definition of the abstract and concrete syntax of the language, the workbench is also used to define transformations that constitute the language's dynamic semantics.

2 Main Purpose

Athos is ultimately intended to support a wide range of optimisation-related traffic scenarios. Up to now, we have implemented support for scenarios in which agents seek to get from a given starting point in a road network with congestion dependant travel times to a predefined destination. For this, they seek to find the route that requires the least amount of time or alternatively try to satisfy a different objective. Since the current traffic situation determines the time it takes to travel roads, each agent is confronted with a dynamic optimisation problem.

Traffic-related problems necessitate the definition of some kind of network. Athos allows the specification of road networks in a straightforward manner. In order to populate the network with agents, nodes can be defined as sources from which new agents originate. Users can specify time-based patterns in which agents are created. It is also possible to model distribution functions to control how source nodes spawn agents with different properties into the system.

Agents differ in two major properties. Firstly, agents can be assigned different *routing modes* that determine how they move inside the network. Most of the time, modellers will assign an a priori destination to agents. However, in order to generate background noise or to simulate public transport routes, it is also possible to define agents that visit a pre-defined sequence of nodes. Secondly, agents can be assigned a *congestion factor*. The congestion factor determines to what extent the respective agent will congest, i.e. slow down, traffic on the road it travels on. This way, agents can represent different types of vehicles. An agent with a high congestion factor could represent a bus or a tractor whereas an agent with a low congestion factor could represent a fast car or a motorcycle.

As was already stated, agents try to get to a pre-defined destination in the least possible amount of time. They calculate the fastest path from their current location to their destination by means of Dijkstra's algorithm. They do so every time they enter a node of the network. Whenever agents recalculate the fastest

[1] https://ccl.northwestern.edu/netlogo/.
[2] https://repast.github.io/download.html.
[3] https://www.eclipse.org/Xtext/.

path, they consider the current traffic situation. For this, roads must be assigned a numerical value that represents the amount of time cars need to travel them. To this end, Athos allows the definition of cost functions as ordinary mathematical expressions. Within these expressions, various properties like the length of the road or the accumulated congestion factor of all vehicles can be used. The value of the accumulated congestion factor depends on both the number and the types of agents that are on the respective road at a given point in time.

The described language features allow for the definition of traffic-dependant travel durations. The higher the accumulated congestion factors of all vehicles on a given road, the longer it takes these vehicles to get to the next road. This, in turn, increases the time window in which other vehicles can further increase the accumulated congestion factor on this road. This way, the language allows for the definition of scenarios where increased numbers of agents congest certain roads in a way that traffic may ultimately grind to a halt. By default, all simulation scenarios track the total amount of time cars have spent in the system.

3 Demonstration

We first consider a network of roads with a single source and a single destination. Agents (i.e. cars) are created in the source node and use a shortest route to the destination while travel time depends on congestion effects on the roads represented by the edges of the network. The following network corresponds to Samuelson's example [3] in which the Braess paradox and the effect of modified cost functions on the choice of routes is examined.

```
1  model Model1 world xmax 60 xmin 0 ymax 60 ymin 0
2  functions // functions used in the model
3  durationFunction traveltime  length + cfactor * accCongestionFactor
4  network
5  nodes // nodes of the network
6    node a (2.0, 15.0)
7    node b (10.0, 15.0)
8    node c (20.0, 15.0)
9    node d0 (30.0, 15.0)
10   node t1 (8.0,25.0)
11   node t2 (15.0, 25.0)
12   node d1 (15.0, 5.0)
13   node d2 (23.0, 5.0)
14 edges // roads that connect the nodes
15   edge ab from a to b length 2.0 cfactor 0.02  function traveltime
16   edge cd from c to d0 length 2.0 cfactor 0.02 function traveltime
17   edge at1 from a to t1 length 10.0 cfactor 0.002 path "ac" function traveltime
18   edge t1t2 from t1 to t2 length 10.0 cfactor 0.002 path "ac"  function traveltime
19   edge t2c from t2 to c length 5.0 cfactor 0.002 path "ac"  function traveltime
20   edge bd1 from b to d1 length 10.0 cfactor 0.002 path "bd"  function traveltime
21   edge d1d2 from d1 to d2 length 10.0 cfactor 0.002 path "bd" function traveltime
22   edge d2d from d2 to d0 length 5.0 cfactor 0.002 path "bd"  function traveltime
23   edge bc from b to c length 1.0 cfactor 0.03 baseSpeed 1.0 function traveltime
24 sources
25    a sprouts (congestionFactor 100.0 destination d0 ) frequency 25.0 every 1 until 3
```

Individual travel time is calculated by means of the duration function *traveltime* defined in line 3. This function accumulates the product of *cfactor* and *congestionFactor* over all individuals using the edge at the very moment and adds it to the *length* of the edge. Note that *congestionFactor* is an attribute of the travelling agent that determines how strong this individual agent will add to congestion while *cfactor* is an attribute of the edge and defines the (linear) congestion effect on this edge. This PIM can either be generated to be run as a NetLogo or a Repast Simphony simulation. While the first offers the easy to use NetLogo system, the latter is created to be run in the scalable Java-based Repast

Simphony system. Note how this model is also computation independent: we just define the problem without any details on how the solution is to be computed.

We will discuss some additional language features by extending the model from above. For example, different types of agents with individual behaviour in how they determine their preferred route can be defined. This can be done very intuitively by definition of an additional type of agent *ecoDriver* and an alternative function *ecoDriverFunction* to compute travel costs relevant to this type of agent. *ecoDriver* agents possess the attribute *fuelConsumption* which is also available to the newly defined function.

```
...
agentAttributes fuelConsumption
agentTypes
agentType ecoDriver congestionFactor 50.0 attr fuelConsumption = 10.0 destination d_ optimises ecoDriverFunction
functions // functions used in the model
durationFunction traveltime  length + cfactor * accCongestionFactor
agentFunction ecoDriverFunction length * fuelConsumption
...
```

Furthermore, we can define two sources (a and b) that generate agents heading towards the same destination d. This is achieved by replacing the last line of the model code by:

```
b sprouts (congestionFactor 100.0 destination d0 ) frequency 50.0 every 1 until 1
a sprouts (ecoDriver probability 95) , (congestionFactor 100.0 destination d0 probability 5) frequency 50.0 every 1 until 1
```

This increases the number of lines of code in the Athos model to 30 and to 587 in the generated NetLogo source.

4 Conclusion

Athos is a DSL that allows for declarative specification of multi-agent traffic simulations. Athos models are generally computationally independent, though users may specify computational details if desired. In any case, models are platform independent. Recently, Athos has been extended to support scenarios that include dynamic travelling salesman problems. We ultimately aim at a language in which a wide variety of traffic optimisation problems can be defined and solved.

References

1. Borenstein, D.B.: Nanoverse: a constraints-based declarative framework for rapid agent-based modeling. In: Yilmaz, L. (ed.) Proceedings of the 2015 Winter Simulation Conference, pp. 206–217. IEEE, Piscataway (2015)
2. Kardoš, M., Drozdová, M.: Analytical method of CIM to PIM transformation in model driven architecture (MDA). J. Inf. Org. Sci. **34**(1), 89–99 (2010)
3. Samuelson, P.A.: Tragedy of the open road: avoiding paradox by use of regulated public utilities that charge corrected knightian tolls. J. Int. Comp. Econ. **1**(1), 3–12 (1992)
4. Sansores, C., Pavón, J.: Agent-based simulation replication: a model driven architecture approach. In: Gelbukh, A., de Albornoz, Á., Terashima-Marín, H. (eds.) MICAI 2005. LNCS (LNAI), vol. 3789, pp. 244–253. Springer, Heidelberg (2005). https://doi.org/10.1007/11579427_25

Check for updates

MASEV: A MAS for the Analysis of Electric Vehicle Charging Stations Location

J. Jordán, J. Palanca(✉), E. del Val, V. Julian, and V. Botti

Universitat Politècnica de València, Camino de Vera s/n, Valencia, Spain
{jjordan,jpalanca,edelval,vinglada,vbotti}@dsic.upv.es

Abstract. Electric vehicles are already a reality in many cities around the world. However, the penetration of the electric vehicle depends on several factors, including the existence of an adequate network of charging stations. This article aims to address this problem by defining a multi-agent system (MAS) that collects information about the state of a city and proposes different configurations for the location of charging stations. More specifically, the proposed system integrates information from heterogeneous data sources and applies a genetic algorithm to characterize the areas where charging stations could potentially be located. The system has been tested with real information from the city of Valencia (Spain).

1 Introduction

In the last years, the Electric Vehicle (EV) is being considered as one of the alternatives to reduce the rate of pollution in urban areas. This situation is encouraging the use of these vehicles by government entities and therefore there is an emerging and important market around them that provides these vehicles to the general public [3].

One of the relevant points to incorporate the electric vehicles to the urban environments is the provision of infrastructures that support the electric charging needs of the vehicles [2]. It is important to consider requirements such as the electrical power at the electricity grid where the stations are located. Another aspect to take into account is that the deployment of the necessary infrastructure is expensive. For this reason, it is important to plan the location of the charging stations that tries to maximize their performance and offers an optimal service to the user of the electric vehicle.

In this paper, we deal with the problem of optimizing the placement of new infrastructures for EV charging. We have designed a multi-agent system that is responsible for gathering and integrating data from heterogeneous sources to characterize a set of potential placements of the charging stations. Based on this information, the system analyzes a set of configurations for the location of

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 326–330, 2018.
https://doi.org/10.1007/978-3-319-94580-4_31

charging stations. Specifically, the agents use a genetic algorithm that considering the data provided about the potential locations (i.e., traffic or population in the area), problem constraints (i.e., maximum number of piles per station or supply of electricity for stations in the city), and a configurable utility function, looks for the best configuration that optimizes the utility function.

2 Main Purpose

The MAS proposed in this work is responsible for the evaluation of a set of configurations for the distribution of charging stations in a city. The MAS integrates a genetic algorithm that evaluates the suitability of each configuration according to a utility function. The proposed MAS consists of a set of agents that offer their functionality through services providing flexibility, scalability, and reusability in different cities. In the following paragraphs, we describe the main features of each agent that participates in the system.

- *Urban Agent* is responsible for analyzing the information about the population of a certain urban area (i.e., neighborhoods or blocks of the city) based on census sources (i.e., census sections of Open Data portals).
- *Traffic Agent* is in charge of the collection of traffic information. Considering this information, the agent is able to determine how much traffic there is on average in a certain defined area of the city.
- *PoI Agent* is responsible for detecting and classifying Points of Interest (PoIs) that could be suitable for the location of charging stations in a city. The agent considers the urban development plan and type of the areas to determine the PoIs. In addition, the agent performs a clustering analysis to detect and remove PoIs that are close to each other. The result of this process is a set of PoIs with a minimum influence area.
- *Popularity Agent* determines the popularity of a PoI taken into account the number of people who visit it and the time those people spend in the area. To obtain this information, the popularity agent uses third party services. One of these services is Google's service, that can be used to retrieve the coordinates of areas of interest. Then, the agent uses the results of the search engine to obtain the estimated time spent by visitors in that area.
- *Social Networks Agent* is responsible for the retrieval of geolocated information from online social networks such as Twitter or Instagram. Based on this information, the agent calculates a popularity value associated to each PoI. The popularity value is based on the activity that occurs through social networks in that area.
- *Data Processing Agent* aggregates all the information obtained by the above mentioned agents. The Data Processing Agent is in charge of combining and completing all the information required to send it to the Emplacement Optimizer Agent.

– *Emplacement Optimizer Agent* is in charge of applying a genetic algorithm to determine the best configuration of charging stations according to the utility function. This agent considers the information provided by the Data Processing Agent as input for the genetic algorithm.
– *User Interface Agent* provides a dashboard that will show the available configuration criteria to be considered in the optimization process as well as the set of information sources that are going to be taken into account by the Emplacement Optimizer Agent. In addition, this agent provides a visual representation of the most suitable configuration of charging stations.

The whole process to obtain a suitable configuration of charging stations in a city consists of five steps: the selection and characterization of PoIs; transforming those PoIs into a Voronoi diagram; the information extraction about the city; the analysis of a set of possible configurations through a genetic algorithm; and the visualization of the results.

3 Demonstration

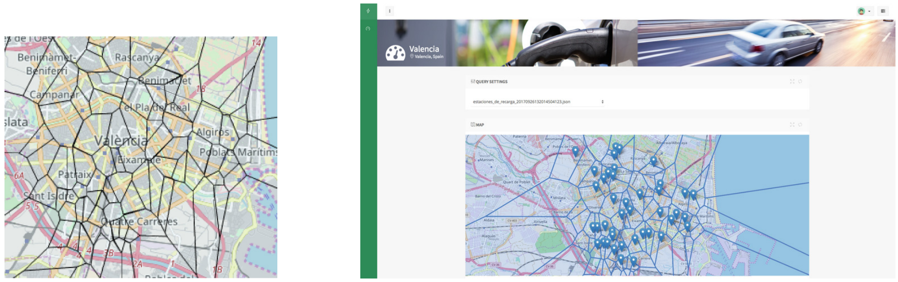
This demonstrator was applied in the city of Valencia (Spain). To feed the genetic algorithm the application uses data retrieved from the Open Data portal supported by the city council. Although in this work the demonstrator has been applied to the city of Valencia, any other city can be added without modification. The application's goal is to optimize the location of Electric Vehicle charge stations throughout the city. According to [1], there are currently 76 charging points in the province of Valencia, while 24 of these are located in the city of Valencia. Some initiatives carried out by the Valencia City Council are intended to improve the city infrastructures to enable the deployment of EVs. One of these initiatives tries to solve the problem of locating the EV charging points where they are really needed, since the installation of these stations is costly and requires an adequate previous electrical infrastructure.

The demonstrator is initialized with a set of potential points of interest (PoIs) called P. This set takes into account the existing charging stations in the city if known. To create this initial set of PoIs for the location of a charging station s_i, we also take into account the GUDP[1].

In each generation each chromosome represents the active PoIs where a station would be installed. Then, the fitness function determines the area of influence of each active PoI by building a Voronoi diagram (Fig. 1a). Next, in a second phase, each polygon of the Voronoi diagram is annotated with the aggregated information collected by the MAS (traffic information, population, social networks activity, etc.).

Taking into account all of these possible points, the system determines the area of influence of each of the PoIs. To do this, a Voronoi diagram is created around the selected points.

[1] General Urban Development Plan.



(a) Voronoi diagram from the candidate points

(b) Configuration of the location of the charging stations displayed by the *User Interface Agent*

Fig. 1. User interface and Voronoi diagrams

The algorithm ends when the *Emplacement Optimizer Agent* returns a set of possible solutions. These solutions have been ordered according to their fitness value, so the best individual in the population is selected as the proposed solution. Figure 1b shows the interface of the application and an example of how a solution is visualized in the tool. The figure shows the locations in which each charge point would be located. Clicking on each point you can also see the number of charging points per station and information about each point.

4 Conclusions

Electric vehicles need an adequate network of charging stations to consolidate their use in cities. This work has presented a MAS which main goal is to calculate the most suitable location of electric vehicle charging stations in a city. The proposed approach allows to obtain real-time mobility data from a city. This information, together with electrical information, is relevant for detecting relevant points/areas of the city to place the charging points.

In summary, the main advantages of the system developed are the access and integration of information from different sources as well as the analysis of the most suitable location of charging stations through the use of optimization techniques. The proposed system has been tested in the city of Valencia, where different experiments have been carried out in order to validate the implementation. The results have demonstrated the correct behavior of the system using real input data from the city of Valencia. Other European and Latin American cities are currently being integrated in the system.

Acknowledgments. This work was partially supported by MINECO/FEDER TIN2015-65515-C4-1-R and MOVINDECI project of the Spanish government.

References

1. Electromaps: Electromaps: Puntos de recarga en Valencia (2017). https://www.electromaps.com/puntos-de-recarga/espana/valencia
2. Klabjan, D., Sweda, T.: The nascent industry of electric vehicles. In: Wiley Encyclopedia of Operations Research and Management Science (2011)
3. Wolfram, P., Lutsey, N.: Electric vehicles: literature review of technology costs and carbon emissions. Technical report, The International Council on Clean Transportation (2016)

On the Way of Protecting MANETs Against Security Threats: A Proactive Approach

Roberto Magán-Carrión[1,2(✉)] and Bernabe Dorronsoro[1]

[1] Department of Computer Science and Engineering, University of Cádiz, 11519 Cádiz, Spain
{roberto.magan,bernabe.dorronsoro}@uca.es

[2] Network Engineering & Security Group (NESG), University of Granada, 18071 Granada, Spain

Abstract. MANETs are specially vulnerable against security attacks. For protecting them, security solutions are traditionally addressed by the so-called intrusion detection and response systems. Nevertheless, using reactive (response) solutions, once the attack is detected, rely in long attack mitigation times. In this work, we propose the use of relay node placement techniques as proactive security solutions. Preliminary experimentation demonstrates the suitability and feasibility of such approaches, not only in reducing the attack mitigation time but in replacing detection and response based traditional solutions.

Keywords: Network security · Proactive security · Node placement MANET

1 Introduction

Nowadays, it is a fact that almost everything is connected. Things and people talk each other for a better daily life. Some examples are smart cities with a huge deployment of sensors *e.g.*, for air pollution measuring; on board wearable devices offering location and health services for the people; or maybe drone based networks deployed in rescue and emergency scenarios supporting communications. However, such services have special communication needs that are supplied by the so called ad hoc networks, especially, the mobile ad hoc networks (MANETs).

Due to their special characteristics, MANETs are more vulnerable [1]. For instance, a highly harmful attack is the so called *blackhole* which aims at interrupting communications in the network by dropping all received packets. In order to fight against them, security solutions are traditionally addressed by intrusion detection systems (IDSs) [2] and intrusion response systems (IRSs) [3]. In this way, once the attack is detected a response to mitigate it is launched. However, depending on the context of application, a delayed response could cause a high cost either economical or, event worse, personal.

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 331–334, 2018.
https://doi.org/10.1007/978-3-319-94580-4_32

Certainly, automatic responses are highly recommended [3] to reduce the time gap from the detection to the effective attack mitigation. Though some proposals appear in the literature, it is a fact that effective responses are really leading out by a security analyst. In spite of the presence of effective response solutions, proactive (tolerant) security approaches could avoid the use of detection and reaction (response) security phases that rely in the reduction of the impact of the attack.

As a proof of concept, we present here the application of the DRNS (Dynamical Relay Node placement Solution) [4,5] as a proactive security solution for fighting against *blackhole* attacks in MANETs. The feasibility of the approach is carried out by the corresponding experimentation.

2 Main Purpose

As previously mentioned, there are two main objectives in the present work:

1. The use of relay node placement solutions like DRNS as valid proactive security approaches.
2. To encourage the research community for finding new proactive security solutions instead of the traditionally based detection and response approaches.

3 Demonstration: Proaction or Reaction?

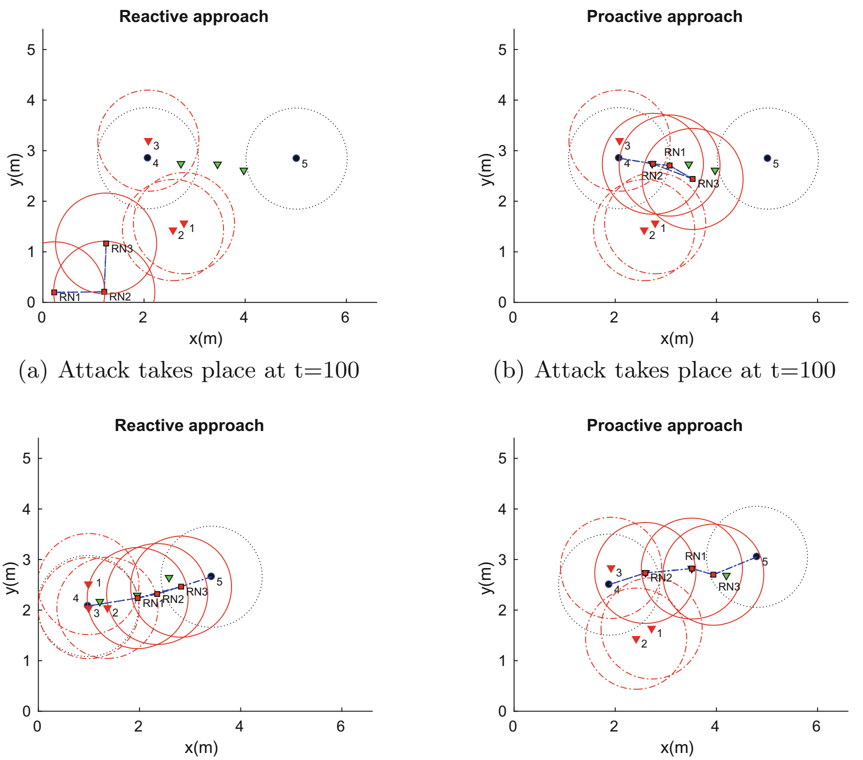
As aforementioned, DRNS is used as a proactive security approach. Briefly, DRNS is in charge of driving the available *relay nodes* (RNs) throughout the network area to optimize connectivity and throughput among the *user nodes* (UNs) over time. The last ones are demanding the network services while the RNs try to guarantee that the first ones receive the best possible network service (see [5] for a detailed explanation).

In this scenario, malicious behaviors (*blackhole*) could degrade the network performance that, depending on the application context, could have severe consequences. To mitigate the undesired effects of such attacks, DRNS is tested for both approaches: reactive (*i.e.*, it is launched when the attack is detected) and proactive (*i.e.*, it is launched at the beginning being active all the time).

3.1 Experimentation and Preliminary Results

A MANET scenario with 5 UNs, three of them will become malicious at attack time ($t = 100$), and 3 RNs is simulated. The last ones are driven by the DRNS engine while the first ones are moved following the reference point group mobility (RPGM) pattern. The coverage radii are set to 1 m to ensure network disconnections typically fall in a predefined area of $6.6\,\text{m} \times 5.4\,\text{m}$. In Fig. 1 we can see the network[1] status at attack and recovery time for reactive (see Figs. 1(a) and (c))

[1] Figure elements legend: UN (blue circle), RN (red square), malicious node (inverted red triangle) and the optimized DRNS computed positions (green inverted triangle).



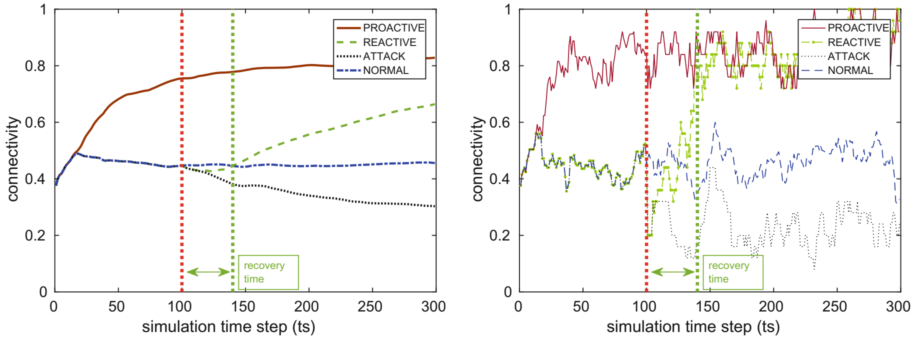
(a) Attack takes place at t=100 (b) Attack takes place at t=100

(c) Connectivity recovered at t=140 (d) Connectivity recovered at t=105

Fig. 1. Network snapshots at different time steps for reactive and proactive security approaches. At top, reactive (a) and proactive (b) solutions at attack time. At bottom, reactive (c) and proactive (d) solutions at recovery time. (Color figure online)

and proactive (see Figs. 1(b) and (d)) DRNS approaches. From the inspection of the figure, the reactive approach takes more time in recovering the connectivity lost. Indeed it is exactly the time gap (around 40 time steps) in driving the relays from their origin locations (left bottom in the figure) to the ones that connect the network. Instead, the proactive approach is continuously locating the RN for maintaining the network connected. This way, as it can be seen in the figure, once the attack takes place the RNs are nearly to connect the network making the recovery time close to zero (around 5 time steps).

Additionally, Fig. 2 shows the connectivity evolution when the proactive (PROACTIVE) and reactive (REACTIVE) approaches are deployed, only with the presence of the attack (ATTACK) and under normal network operation (NORMAL). From the results, on one hand, we also clearly see a longer response time for the reactive solution than the proactive one. On the other hand, we conclude that using the proactive approach, not only mitigates the effect of the attack, but improves the network connectivity over the time.



(a) Cumulative mean (b) Instantaneous

Fig. 2. Network connectivity evolution. Vertical red line means the attack time (t = 100) while its homologous in green depicts the recovery time (t = 140) for the reactive approach. (Color figure online)

4 Conclusions

In this work we demonstrated the suitability of RN node placement systems as proactive solutions for network security. Moreover, we also tested the convenience of using proactive instead of reactive security solutions, especially for reducing the attack mitigation time. Finally, we are pretty convinced in the use of proactive (tolerant) solutions as a feasible alternatives to the traditionally based detection and reaction (response) security approaches though much work have to be done in this way.

Acknowledgment. This work has been partially supported by Spanish Government-MINECO (Ministerio de Economía y Competitividad) and FEDER funds, through projects TIN2014-60346-R and TIN2014-60844-R.

References

1. Sánchez-Casado, L., Magán-Carrión, R., García-Teodoro, P., Díaz-Verdejo, J.E.: Defenses against packet dropping attacks in wireless multihop ad hoc networks. In: Khan, S., Lloret Mauri, J. (eds.) Security for Multihop Wireless Networks, pp. 377–400. CRC Press, Boca Raton (2014)
2. Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K.: Network anomaly detection: methods, systems and tools. IEEE Commun. Surv. Tutor. **16**(1), 303–336 (2014)
3. Inayat, Z., Gani, A., Anuar, N.B., Khan, M.K., Anwar, S.: Intrusion response systems: foundations, design, and challenges. J. Netw. Comput. Appl. **62**, 53–74 (2016)
4. Magán-Carrión, R., Camacho, J., García-Teodoro, P., Flushing, E.F., Di Caro, G.A.: DRNS: dynamical relay node placement solution. In: Demazeau, Y., Ito, T., Bajo, J., Escalona, M.J. (eds.) PAAMS 2016. LNCS (LNAI), vol. 9662, pp. 273–276. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39324-7_29
5. Magán-Carrión, R., Camacho, J., García-Teodoro, P., Flushing, E.F., Di Caro, G.A.: A dynamical relay node placement solution for MANETs. Comput. Commun. **114**(C), 36–50 (2017)

Coordinated UAV Search and Rescue Application with JaCaMo

Marcelo S. Menegol(✉), Jomi F. Hübner, and Leandro B. Becker

Universidade Federal de Santa Catarina, Florianópolis, SC, Brazil
m.s.menegol@posgrad.ufsc.br, {jomi.hubner,leandro.becker}@ufsc.br

Abstract. This paper presents an agent-hardware integration architecture used to embed JaCaMo agents on Unmanned Aerial Vehicles (UAVs). A search and rescue application was developed to experiment with these UAVs. The JaCaMo programming platform and its Moise organizational support are used to provide organization-based coordination capabilities to the application.

Keywords: JaCaMo · Coordination · UAV · Embedded systems

1 Introduction

The use of unmanned aerial vehicles (UAVs) has increased in popularity over the last years. These mobile robots have been employed by many industries, from military to entertainment. One of the possible applications of these UAVs is on search and rescue operations. It consists of looking for victims in an area (e.g. people drowning on the water) and delivering supplies to them (e.g. a life-buoy).

Search and rescue applications can benefit from the use of a heterogeneous team of UAVs. To search an area and swiftly deliver supplies to the victims, a single UAV carrying both camera and supplies can be used. But the distance that an UAV can travel is inversely proportional to the weight it carries. Instead, two UAVs, one with a camera and another just to deliver resources, can provide a more efficient solution. A larger area can be covered if the scout UAV searches for victims and the courier UAV only flies directly towards the delivery locations. In this proposal, the UAVs have to *coordinate* properly to fulfill their mission.

This paper describes the agent-hardware integration architecture used to perform a simulated search and rescue operation with UAVs using JaCaMo, a multi-agent development framework [1]. The coordination required by the application is supported by Moise [7], an organization-based coordination language provided by JaCaMo. The application was developed to be compatible with a standard UAV flight controller board but, for the purpose of this demonstration, is executed on a simulated environment.

The authors are grateful for the support given by CAPES (Pró-Alertas project).

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 335–338, 2018.
https://doi.org/10.1007/978-3-319-94580-4_33

2 Main Purpose

This demonstration aims to show that it is possible to develop embedded applications that require coordination with JaCaMo. Moreover, it shows the basic requirements that an agent-hardware integration architecture should have to allow the use of agent-oriented approaches on these systems.

More specifically, it demonstrates that the agent-oriented approach can be used to solve a relevant problem: autonomous search and rescue missions with autonomous UAVs. JaCaMo provides easy programming tools to expand the scope of applications where those aircrafts can be used on.

3 Demonstration

The application consists of having two UAVs that cooperate in order to rescue cast-away victims on waters. One UAV (scout) searches for survivor, using a camera, and the other (courier) delivers life-buoys to them when they are found. Operating this way, the UAVs could cover a bigger area than if they had redundant equipment (i.e. both carrying camera and buoy) and, therefore, had to carry more weight.

For this application, software agents execute on a single-board computer (BeagleBone Black [4]) that goes embedded on the UAV. The agent then acquires information from the flight controller board (such as position of the aircraft), reasons based on its current situation, and commands the controller to move the aircraft. Moreover, since the agents also need to coordinate themselves, inter-agent wireless communication is also provided.

Agent-oriented frameworks do not usually provide interfaces for specific devices such as UAVs flight controller boards. The JaCaMo framework allows for its agents' architecture to be customized though. By overwriting a few methods in Java we can modify how the agent perceives its environment, act upon it and communicate with other agents [2]. In this case, the UAV's flight controller exchanges data via a specific protocol (MAVLINK [5]).

An agent-hardware integration middleware was developed. It has a layered structure: a *head*, responsible for the agent's customization, and a *body*, responsible for properly interfacing with other devices. Both parts are connected via a socket interface, a resource that is available in many programming languages. The *head* is composed of two parts: a Java class that customizes the agent's architecture (*cortex*) and another Java class just to handle the socket connection (*bulb*), launched by the *cortex* as a thread. The *body* is more loosely defined as a program that sends percepts to, executes actions received from, and properly dispatches messages to and from the *head*. For the purpose of this application, it was developed in Python so that the DRONEKIT library could be used to interface with the flight controller board via MAVLINK. Figure 1 represents the concept: various UAVs, each one with the whole architecture embedded, using a wireless device for communication and coordination. A video of a single UAV flying with this architecture embedded on it can be found at https://youtu.be/FcS4QDtrBCI.

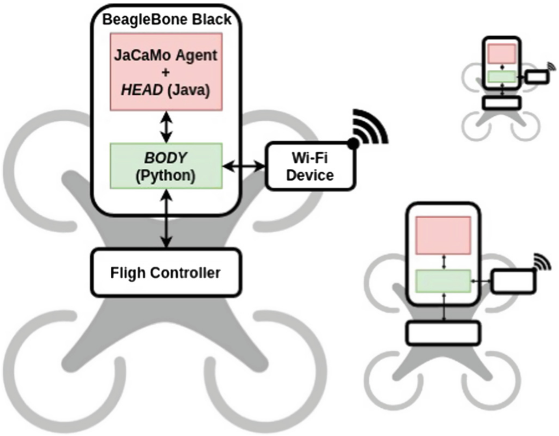


Fig. 1. General view of the application: agents and UAVs.

For the purpose of this demonstration, a simulator [3] was used, one instance for each agent. Moreover, a program to handle the interface with the wireless device was developed. It is launched as a thread of the *body*. Another Python program that simulates an ad-hoc wireless network, in a mesh topography, was also developed. All agents are connected to this same message dispatcher. The whole application for a single UAV is shown on Fig. 2. For simulation purposes, the other agents were executed on a personal computer. The full solution can be found at https://github.com/msmenegol/UAVExperiments.

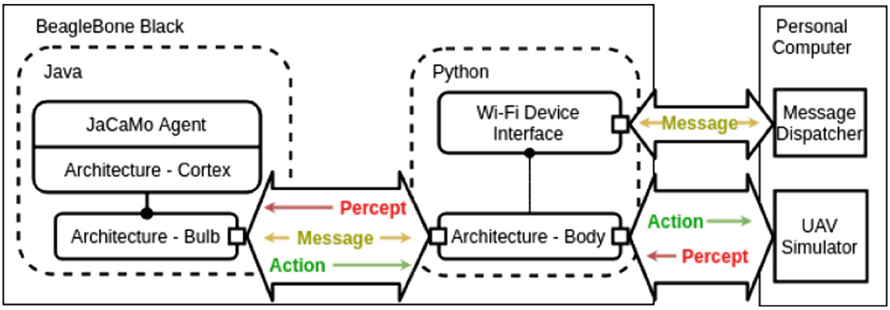


Fig. 2. Overview of the agent-hardware integration architecture.

JaCaMo uses Moise notion of schemes to enforce coordination. Briefly, a scheme is a set of global goals decomposed into sub-goals assignable to agents. Agents have thus a mission (a set of goals) when participating in the scheme execution. To initially start the application and distribute the missions to each agent, a manager agent is used.

The execution starts with both agents loading the same organizational specification. Initially, each one commits to a different mission (as assigned by the manager agent). When they share their scheme, each recognize that both missions (scout and courier) have agents committed to them. The scout then starts by taking off and flying towards the pre-defined waypoints of its route. The courier registers that it has a buoy and that no victims have been found yet, and waits on the ground. Whenever the scout gets close enough to any victim it sends a message to the courier with the victim's location. The courier then takes off and goes directly to where the victim is and delivers a buoy.

As the courier returns from the delivery, the scout finds two other victims, one shortly followed by another, and warns the courier. The courier goes back to its landing spot, lands, get recharged with one buoy, and goes to the second victim. While it is delivering, the scout reaches its final waypoint and returns to its landing spot, finishing its mission. Through the common scheme, the courier finds out the state of the scout's missions and enters the final stage of its own mission: when the number of rescued victims matches the amount of victims found, it can go home and finish as well. The courier proceeds to perform its remaining deliveries and returns home, finishing its missions and accomplishing the global goal.

4 Conclusion

This paper briefly shows an integration architecture that enables the use of JaCaMo agents on embedded systems. It allows for a variety of ready-made libraries and modules to be used on the *body* part of the architecture, making the development process of many applications easier. With this, a wider range of applications can be explored with JaCaMo.

References

1. Boissier, O., Bordini, R., Hübner, J.F., Ricci, A., Santi, A.: Multi-agent oriented programming with JaCaMo. Sci. Comput. Program. **78**(6), 747–761 (2013)
2. Bordini, R.H., Hübner, J.F., Wooldridge, M.: Programming Multi-Agent Systems in AgentSpeak Using Jason. Wiley, Hoboken (2007)
3. ArduPilot Development Team: SITL simulator, January 2018. http://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html
4. Coley, G., Kridner, J.: BeagleBone Black System Reference Manual, January 2018. https://github.com/beagleboard/beaglebone-black/wiki/System-Reference-Manual
5. Dronecode Project: MAVLink Micro Air Vehicle Communication Protocol, January 2018. http://qgroundcontrol.org/mavlink
6. 3D Robotics: DroneKit-Python's documentation, January 2018. http://python.dronekit.io
7. Hübner, J.F., Boissier, O., Kitio, R., Ricci, A.: Instrumenting multi-agent organisations with organisational artifacts and agents. J. Auton. Agents Multi-Agent Syst. **20**(3), 369–400 (2010)

AgentOil: A Multiagent-Based Simulation of the Drilling Process in Oilfields

Yazan Mualla[1,2](✉), Robin Vanet[2], Amro Najjar[3], Olivier Boissier[3], and Stéphane Galland[1]

[1] LE2I, Univ. Bourgogne Franche-Comté, UTBM, 90010 Belfort, France
{yazan.mualla,stephane.galland}@utbm.fr
[2] Univ. Lyon, IMT Mines Saint-Étienne, 42023 Saint-Étienne, France
robin.vanet@telecom-st-etienne.fr
[3] Univ. Lyon, IMT Mines Saint-Étienne, CNRS, Laboratoire Hubert Curien UMR 5516, 42023 Saint-Étienne, France
{amro.najjar,olivier.boissier}@emse.fr

1 Introduction

Oil&Gas have become the world's most important source of energy since the mid-1950's. For instance; Britain oilfields produce each year about 76 million tonnes of oil equivalent. This provides 76% of the UK's total primary energy [5]. In oilfields wells, a drilling rig is used to create a bore-hole in the earth's subsurface with a Bottom Hole Assembly (BHA), which is a composition of several drilling tools with various functionalities, searching for natural resources.

While drilling, the increasing temperature damages the tools. To monitor the status of tools, sensors are attached to them in order to read useful data like temperature. Read data are sent up-hole to the field engineer (several kilometers above the down-hole tools). When deemed necessary, the latter activates the drilling mud coolers to reduce the temperature of down-hole tools by pumping cooled mud down-hole. However, this process suffers from four flaws: (i) Existing wells rely on analogue telemetry that has long response time and unreliable signal. (ii) The effectiveness of the mud cooler may become inefficient in high temperature conditions. (iii) The process relies heavily on constant human monitoring and intervention. This makes it error-prone, unresponsive, and fault-intolerant. (iv) The current decision-making process is highly centralized up-hole.

To overcome these flaws, we propose a spec-driven multiagent system (MAS) for mitigating high temperature by distributing the monitoring and control process between different agents that coordinate with each other through voting rules. To demonstrate our proposal, we present AgentOil, a multiagent-based simulation tool to simulate the drilling process in oilfields.

2 Main Purpose

Most of the works in the MAS domain in Oil&Gas industry are still theoretical, and the concentration is mainly on supply chain and management aspects, while

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 339–343, 2018.
https://doi.org/10.1007/978-3-319-94580-4_34

the drilling process is not addressed. In order to implement the proposed system properly, we rely on the literature of drilling mechanics and geothermal equations. In particular, we consider the following key parameters[1]. Measured Depth (MD) is the length of the well bore-hole, while True Vertical Depth (TVD) is the vertical distance from the surface until the lowest drilling tool (the bit). Inclination (drilling angle) is the deviation from vertical that relates the MD with TVD using the Pythagorean equation. The rate of increase in temperature per unit depth in Earth is called Geothermal Gradient. The estimation of the down-hole temperature of drilling tools is shown in Eq. 1.

$$DownHoleTemp = SurfaceTemp + \mathrm{TVD} * GeothermalGradient \tag{1}$$

Drilling parameters are used to control the drilling process. Mainly, we focus on three essential parameters: Force (Weight on bit or WOB), Rotation (Revolutions per minute or RPM) and Flow rate of drilling mud. The speed of drilling process (Rate of penetration or ROP) is shown in Eq. 2 [4].

$$ROP = K\frac{\overline{WOB}^{K}}{a^{p}}r \tag{2}$$

Where K: Formation factor, is a constant related to a given earth formation hardness; $\overline{WOB}$: function of WOB; r: function of RPM; a^{P}: function of the flow rate and the bit characteristics (c.f. [4] for more details).

Within AgentOil, the down-hole tools are considered as agents and enhanced with a coordination mechanism to mitigate high temperature autonomously in real-time by controlling a down-hole actuator through a voting process, where the vote of each tool is determined from its temperature specification. In order to aggregate the votes, AgentOil implements three well-known voting rules in the MAS domain [1]: Plurality, Borda count and Condorcet.

When the temperature of one tool exceeds its specification limit, voting is triggered by its agent. Each tool agent votes to start the down-hole actuator, and an overall decision is aggregated. Starting the actuator reduces the drilling speed, hence it delays the time to reach higher temperature. Although this leads to slower drilling, it mitigates the temperature raise, thereby save the tools from failure allowing them to drill deeper. To minimize the drilling speed reduction, the actuator is not activated if there is no need. The whole process is done in real time without the intervention of up-hole entities.

3 Demonstration

AgentOil (Fig. 1) is implemented using RePast simphony [2], a multiagent simulation environment, as it has significant operational and executional features [3]. A simulation run represents a whole drilling run in real-life drilling scenario where the BHA is made at surface to drill to a specific depth (a.k.a. Total Depth) before the need to go up to surface to change it with new tools. Table 1

[1] The first author has five years of experience in Oil&Gas industry.

Table 1. Simulation initial parameters

Parameter	Default value	Description
Number of agents	5	Number of tool agents in the simulation
Voting rule choice (1: Plurality, 2: Condorcet, 3: Borda count)	1	Voting rule used in the voting system for this simulation run
Starting run depth (m)	4500	Measured depth at the start of the run
Total Depth (m)	5000	Measured depth at the end of the run
Drilling angle (degree)	30	Bore-hole angle compared to vertical line
Hole diameter (inch)	6	Diameter of the bore-hole
Initial RPM (0–100)	100	Initial RPM set by the user
Surface temperature (degC)	20	Temperature at surface
Geothermal Gradient (degC/m)	0.03	Temperature incremental rate per meter
Initial WOB (0–30 klbf)	1	Initial WOB set by the user
WOB increase per tick (klbf)	0.05	Amount of constant increase in WOB

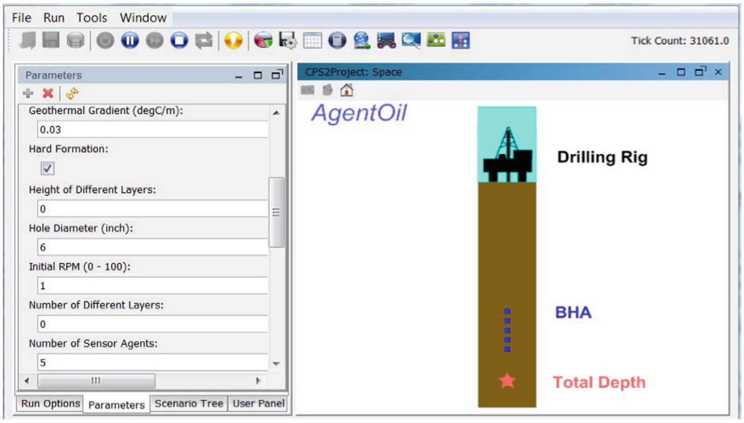


Fig. 1. AgentOil snapshot: initial parameters (Left), 2D drilling visualization (Right)

lists some of the initial parameters[2] that the user can change before starting. Once the simulation starts, all tool agents are created and set accordingly in the environment. A simulation tick corresponds to one minute. Thus, the results

[2] Full list is not provided due to space limitations.

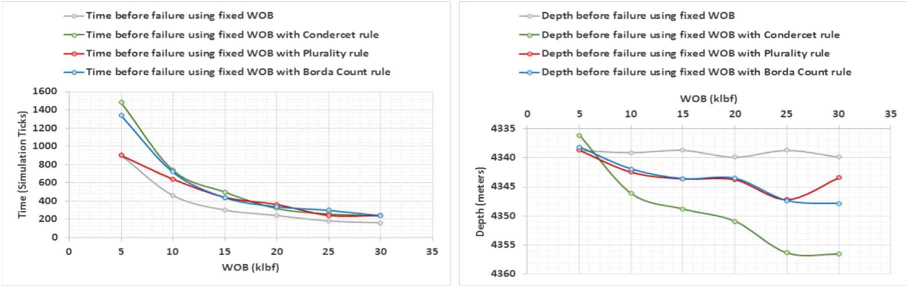


Fig. 2. Gain in time and depth with the proposed system (Color figure online)

are normalized as speed is given in m/min. In each tick, drilling parameters are updated to calculate the ROP (Eq. 2). Therefore, the BHA goes down the hole, and temperature increases with depth (Eq. 1).

The ultimate goal is to strike a balance between maintaining a high drilling speed and guaranteeing the tools integrity by controlling the down-hole actuator. Each agent is aware of the temperature specification of its tool, and decides whether to start a voting cycle accordingly. A simulation run can end either successfully by reaching Total Depth if the mitigation measure down-hole is sufficient or unsuccessfully with a tool failure before reaching Total Depth. Additionally, a complete log of the simulation run is provided in both cases.

AgentOil allows to simulate a varied number of drilling tools, and in the following evaluation we use five tools to adhere to most common real life scenarios. Figure 2 shows the results in terms of time and depth of running AgentOil tens of times with the proposed system (colored curves) and without it (gray curve) for different values of fixed WOB (x-axis). Figure 2 (left) illustrates the time elapsed before failure (how much time the tools survived in high temperature environment). Correspondingly, Fig. 2 (right) plots the drilled depth before failure. In both figures, three voting rules were examined: Condorcet (green), Plurality (red), Borda count (blue). From Fig. 2 (left), it can be noticed that all colored curves are above the gray curve, which means that all runs of the proposed system (with different voting rules) are better than runs without enabling the proposed system. Figure 2 (right) shows that the colored curves are below the gray curve (more drilled depth), significantly with high WOB.

4 Conclusions

AgentOil provides the ability to control the most important drilling parameters and simulate drilling runs in different hole sizes in drilling sections with the necessary information of each drilling run. We used AgentOil to implement the proposed multiagent system and produce results. The results show that the down-hole tools agents are capable of mitigating high temperature down-hole autonomously in real-time by socially reacting to the increase in temperature. This increases the life cycle of down-hole tools allowing them to drill deeper.

References

1. Brandt, F., Conitzer, V., Endriss, U.: Computational social choice. In: Multiagent Systems, pp. 213–283 (2012)
2. Collier, N.: Repast: an extensible framework for agent simulation, vol. 36. The University of Chicago's Social Science Research (2003)
3. Mualla, Y., Bai, W., Galland, S., Nicolle, C.: Comparison of agent-based simulation frameworks for unmanned aerial transportation applications. Procedia Comput. Sci. **130**, 791–796 (2018). (In: 7th International Workshop on Agent-based Mobility, Traffic and Transportation Models, Methodologies and Applications)
4. Osgouei, R.E., Özbayoğlu, M.: Rate of penetration estimation model for directional and horizontal wells. In: 16th International Petroleum and Natural Gas Congress and Exhibition of Turkey (2007)
5. The UK Oil & Gas Industry Association: Oil & Gas UK Economic Report 2017, ISBN 1903004942. Technical report, OIL&GAS^{UK} (2017)

A Social Robot Assisting in Cognitive Stimulation Therapy

Esther Salichs(✉), Enrique Fernández-Rodicio, José Carlos Castillo, Álvaro Castro-González, María Malfaz, and Miguel Ángel Salichs

Department of Systems Engineering and Automation, University Carlos III of Madrid, Getafe, Spain
{esalichs,enrifern,jocastil,acgonzal,mmalfaz,salichs}@ing.uc3m.es

Abstract. In recent years, the demand for caregivers and therapists has raised together with the life expectancy. Elder people demand adapted environments and specialized care that allows for an active ageing. In order to assist in their daily tasks, social robots appear as a complement to caregivers. In this demo we introduce a robot capable to conduct cognitive stimulation sessions, complementing the therapists or even conducting the sessions autonomously. This robot has been designed to work both at home and daycare facilities.

Keywords: Cognitive stimulation therapy · Social robot
Human-robot interaction

1 Introduction

Nowadays, there are a number of elders in need of personal assistance and care. In these cases, the presence of caregivers plays an essential role to improve their life quality. But for the caregivers, being responsible for another person 24/7 becomes in some cases a stressful task, both physically and emotionally. Robots can also help to mitigate this situation, taking care of the patient well-being and reducing the workload of the caregiver.

More specifically, social robots have been employed in the task of assisting elders and people with cognitive impairment. There are examples such as Paro [1], a baby harp seal robot designed to reduce the levels of stress in patients, to encourage them to socialize, and to improve their motivation. Other relevant examples are Kompai [2] and Care-O-Bot [3], both designed to serve as a pseudo-caregivers, assisting elders in their home chores. Although many works have shown how robots can improve patients' daily life, none of them aims to also assist caregivers in their job. For this purpose, we have developed the social robot Mini [4], a plush-like desktop robot designed to be appealing and create emotional bonds with the users. Although it can be applied in several scenarios (e.g. safety, entertainment or personal assistance, for example), this work presents how Mini is capable of conducting cognitive stimulation exercises.

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 344–347, 2018.
https://doi.org/10.1007/978-3-319-94580-4_35

2 Main Purpose

Traditionally, the care of elders and people with special needs has been trusted to human caregivers although, nowadays, technology is proposing solutions to assist in these tasks. This is the case of cognitive stimulation, where different applications have been designed to assist the caregiver during the stimulation sessions. Among the advantages of incorporating technologic aids to stimulation, we can find the automatic assessment of the user's state along the exercises. Nevertheless, these therapies tend to be monotonous and lack the positive aspects of therapy related to human-to-human interaction. In this sense, robots appear as a solution to these issues providing multimodal interaction, which makes the communication more natural for the users. In contrast to simple computer applications, robots can also interact with the environment, allowing for a range of exercises that involve using physical objects.

One of the limitations of robotic therapies is that these robots lack autonomy, with a therapist or caregiver controlling the evolution of the exercises. In our case, we proposed an autonomous solution in which a social robot could be able to conduct a stimulation session with exercises previously defined by experts. Our goal is to communicate with the elders, addressing the specific challenges arising from the interaction with people with cognitive problems as well as other limitations such as hearing, speech or vision issues.

3 Demonstration

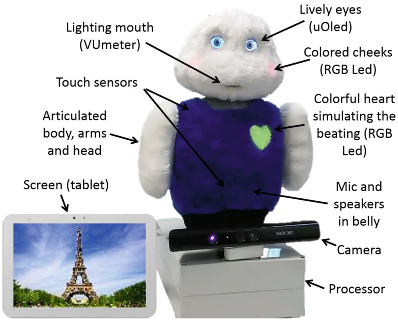
This section provides some insights about the different capabilities that will be shown in the demo. It is also important to describe the multimodal interaction capabilities that our robot can use to interact with people, taking into account some limitations that people with cognitive impairment may present. Interactions have been designed to be one-to-one, with the user sitting in front of the robot as seen in Fig. 1(a).

3.1 Multimodal Interaction

The robot Mini was designed to handle multimodal communication, in order to interact with users in a natural way (see Fig. 1(b)). The robot is covered with a soft material in order to give it a friendly appearance. Regarding the inputs, our robot is equipped with a microphone for speech-based communication, touch sensors in the shoulders and belly that allow the user to interact physically with Mini, an RGB-D camera to extract visual and depth information from the environment, and a tablet that can work both as an input device (through menus) and an output device, displaying photos, videos, music and other external applications.

Mini can also move and perform expressive gestures as it has five degrees of freedom (two in its head, one in each arm and one in the base). Additionally, Mini is equipped with a pair of speakers that output the robot's voice, and multiple

(a)



(b)

Fig. 1. (a)Mini elements. (b)Elder interacting with Mini during a cognitive stimulation session

RGB LEDs on the mouth, cheeks and heart. These LEDs are mainly used to transmit liveliness, although they can also be used to communicate information to the user during the exercises. Finally, Mini's eyes are two uOLED screens that display different GIFs, allowing the robot to look in different directions and change the expression in order to transmit different moods.

3.2 Demonstrating the Robot Capabilities

This demo presents the different capabilities and exercises that have been designed in cognitive stimulation sessions. These cover almost all the main cognitive functions, which are the basis for the psycho-stimulation methods traditionally used by therapists and can be classified into seven categories: (i) *Temporal orientation*, where the robot asks the user about the current date, season, month, etc.; (ii) *Attention*, where the user has to react as quickly as possible, interacting with the screen or the touch sensors, depending on the robot requests; (iii) *Gnosis or perception*, recognizing and identifying different types of auditory and visual stimulus; (iv) *Memory*, including both short-term memory (e.g. remember a list of words), and semantic memory (e.g. general knowledge); (v) *Executive functions* with exercises involving processes related to planning, reasoning, inhibition, decision making, etc.; (vi) *Calculus*, which includes simple calculations, such as counting coins or making simple mathematical operations; And (vii) *language*, where users have to find words related to a category, complete a saying or find the proposed word with missing letters.

The process to execute the demo is as follows: (i) The robot starts asleep, so the user has to wake it up touching the shoulders or belly; (ii) The robot ask the user what he want to do, and he has to answer by voice. Apart of the psycho-stimulation exercises, the robot has the possibility of introducing itself and to present some of its capabilities. If the user does not want to interact with the robot, it will go back to the sleeping state. When the robot finish the demanded task, it comes back to this step and repeat again what the user wants to do. (iii) If the user selects the psycho-stimulation option, the robot asks him if he wants to execute a session or to test a specific exercise. In the case of an exercise, firstly he has to select the cognitive function category, then the exercise to be performed and finally, the level of difficulty; And (iv) when the exercise or the session has finished, the robot repeats the step three. If the user does not want to continue with the cognitive stimulation, then the robot returns to the step two.

4 Conclusions

This demonstration presents a cognitive stimulation session conducted by a robot to help elders with cognitive impairment. This session consists of a series of exercises where the robot asks the user to complete different tasks, and analyses the response obtained from the patient, being also in charge of making appropriate decisions based on the information extracted from the environment in order to advance in the session. Our robot is designed to work alongside a caregiver and can manage multimodal interaction in order to make communication as natural as possible for the user.

Acknowledgments. The research leading to these results has received funding from the projects: Development of social robots to help seniors with cognitive impairment (ROBSEN), funded by the Ministerio de Economia y Competitividad; and RoboCity2030-III-CM, funded by Comunidad de Madrid and cofunded by Structural Funds of the EU.

References

1. Shibata, T., Wada, K.: Robot therapy: a new approach for mental healthcare of the elderly - a mini-review. Gerontology **57**(4), 378–86 (2011)
2. Pieskä, S., Luimula, M., Jauhiainen, J., Spitz, V.: Social service robots in public and private environments. In: Recent Researches in Circuits, Systems, Multimedia and Automatic Control, pp. 190–196 (2012)
3. Graf, B., Reiser, U., Hägele, M., Mauz, K., Klein, P.: Robotic home assistant care-o-bot® 3-product vision and innovation platform. In: 2009 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), pp. 139–144. IEEE (2009)
4. Salichs, E., Castro-González, Á., Malfaz, M., Salichs, M.: Mini: a social assistive robot for people with mild cognitive impairment. In: New Friends, pp. 31–32 (2016)

Check for updates

Swarm of Satellites: Multi-agent Mission Scheduler for Constellation of Earth Remote Sensing Satellites

Petr Skobelev[1(✉)], Elena Simonova[1], Alexey Zhilyaev[1], and Vitaly Travin[2]

[1] Samara National Research University, Samara, Russia
petr.skobelev@gmail.com
[2] Smart Solutions – Knowledge Genesis Group, Samara, Russia

Abstract. The paper describes the multi-agent system for scheduling of actions of a constellation of Earth remote sensing satellites. The functionality and architecture of the system are presented and the key scenarios of event processing are given, which demonstrate the advantages of multi-agent technology.

Keywords: Multi-agent technology · Constellation of satellites
Earth remote sensing · Planning and scheduling

1 Introduction

One of the new trends in Earth remote sensing is the shift from individual management of satellites – to managing swarms of satellites [1].

In swarm of satellites concept the new demands or any other events trigger adaptive re-scheduling of resources and new schedule is self-organized by negotiations of agents of orders, satellites, ground stations, on-board equipment, etc.

In this paper, we will briefly describe the functionality and architecture of the developed system and demonstrate key scenarios of events processing. It will demonstrate advantages of multi-agent technology for real-time resource allocation, scheduling and optimization, including better time-to-go, flexibility and high efficiency of resources.

2 Demonstration of Swarm of Satellites

2.1 Multi-agent Mission Scheduler

The Multi-agent Mission Scheduler is based on developed multi-agent technology for resource allocation, scheduling, optimization and controlling [2].

In the developed approach, agents of orders continuously negotiate with agents of satellites, ground stations and on-board equipment their allocations and pro-actively try

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 348–352, 2018.
https://doi.org/10.1007/978-3-319-94580-4_36

to improve their positions in schedules. Agents take into consideration a number of specific criteria, preferences and constraints – for example, each satellite has a memory limit, so a satellite cannot take images of all available areas and has to choose between image acquisition and downlinking.

Agents of orders take into consideration two criteria – time of getting images and resolution. These two criteria can certainly contradict one another, so the agent uses the convolution of criteria with specified weight coefficients. Agents of resources are trying to improve their load and to be fully utilized.

The resulting schedule is considered as unstable equilibrium achieved on virtual market, which can be easily re-balanced by events if required [3].

2.2 The Main Screens for Demonstration

Initial data, process of adaptive scheduling and its results are displayed on several screens and terminals for entering new events. The screens "Physical world" and "Create Mission" shows the space system configuration and locations of observation areas and ground stations. It can simulate their relative movements on 3D model (Fig. 1).

Fig. 1. Fragment of the screens "Physical world" and "Create Mission"

The screen "Mental world" illustrates the semantic links established between the agents in the process of schedule self-organization. All the agents are shown by circles gradually changing color from red to green until the agent fulfills its objective functions. Once the request is included in the schedule, links are built between the request circle and the selected satellite and ground station circles. If any of the resources have been removed, some of the orders scheduled earlier will no longer be possible. The links will disappear. Unscheduled orders will try to find an open vacant slot on the remaining resources. Finally, the schedule will be re-configured (Fig. 2).

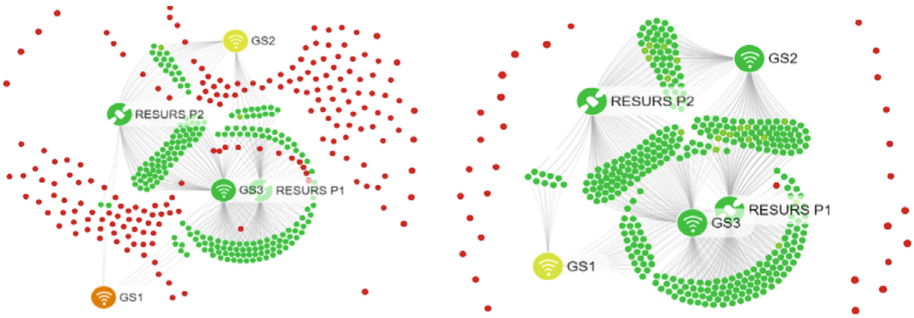


Fig. 2. Fragment of "Mental world" after excluding the satellite and repairing the schedule

The next screen represents a plan of actions of a selected satellite. Here, visibility zones of satellites are shown with all areas and ground stations as well as a plan of executing operations on image acquisition and data transmission (Fig. 3).

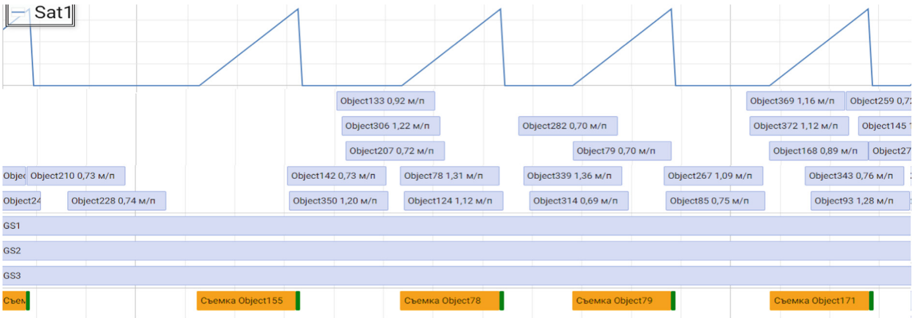


Fig. 3. Fragment of satellite's plan of actions

The diagrams in the next screens show the evolution of the schedule quality in the course of schedule self-organization. The agents create the schedule by iterations. At the initial time point, nothing is scheduled anywhere: no links between agents, full chaos. However, gradually, the order agents negotiate the placements with satellite and ground station agents.

First, agents of orders occupy open slots and then improve positions by making coordinated swaps and shifts in the schedule. When the system is reaching balance of agents interests (consensus) which cannot be improved further in any trials, the system stops and presents the final schedule to the user (Fig. 4).

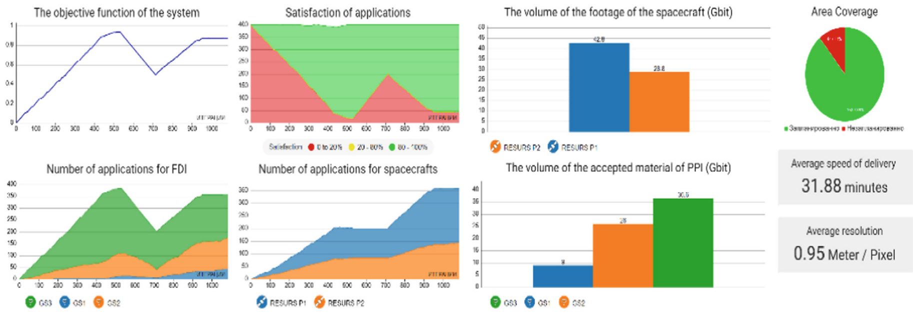


Fig. 4. Screens with results of scheduling

The system recognizes a new order, satellite failure or ground station outage as an external event. Multiple events change the schedule repeatedly, pushing down the agents' objective functions (Fig. 5). However, the agents are able to effectively deal with the changes by adaptive re-scheduling of the resources.

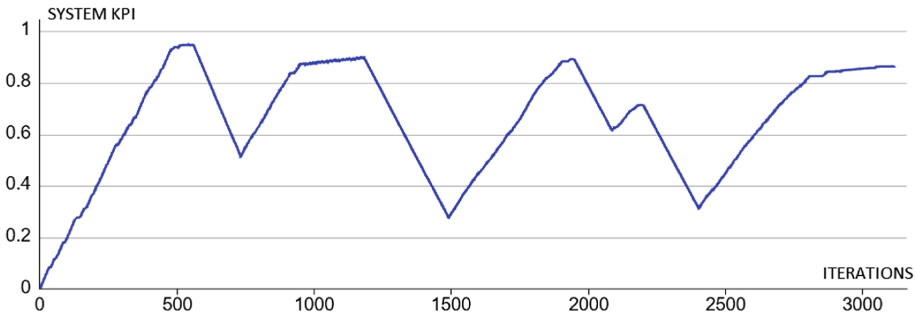


Fig. 5. System KPI during the planning process

As it is shown in Fig. 5, some of the events significantly decrease the system KPIs but then it adaptively re-schedules the resources and improves the situation.

The demo is available on https://youtu.be/JOjhaIRBVdI (short version) and https://www.youtube.com/watch?v=r7vKK9XnTCE (full version).

3 Conclusions

In this paper, we demonstrate the multi-agent system for solving the Earth sensing satellites scheduling problem in dynamic environment. In these circumstances, the system vividly demonstrates its unique adaptability to quickly restore the damaged schedule and to re-schedule the given resources of satellites in real time.

Acknowledgment. This paper was prepared with the financial support of the Ministry of Education and Science of the Russian Federation – contract №14.578.21.0230, project unique ID is RFMEFI57817X0230.

References

1. Belokonov, I., Skobelev, P., Simonova, E., Travin, V., Zhilyaev, A.: Multiagent planning of the network traffic between nanosatellites and ground stations. Procedia Eng. Sci. Technol. Exp. Autom. Space Veh. Small Satell. **108**, 118–130 (2015)
2. Shoham, Y., Leyton-Brown, K.: Multi-Agent Systems: Algorithmic Game Theoretic and Logical Foundations. Cambridge University Press, Cambridge (2009)
3. Skobelev, P., Simonova, E., Zhilyaev, A., Travin, V.: Multi-agent planning of spacecraft group for earth remote sensing. In: Proceedings of the 5th International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing (SOHOMA15), vol. 640, pp. 309–317 (2015)

Demonstration of Tools Control Center for Multi-agent Energy Systems Simulation

Brígida Teixeira[1], Francisco Silva[1], Tiago Pinto[1,2](✉), Gabriel Santos[1], Isabel Praça[1], and Zita Vale[1]

[1] GECAD – Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, Institute of Engineering – Polytechnic of Porto (ISEP/IPP), Porto, Portugal
{bccta,fspsa,gajls,icp,zav}@isep.ipp.pt

[2] BISITE – Research Centre, University of Salamanca, Salamanca, Spain
tpinto@usal.es

1 Introduction

The use of energy from renewable sources is one of the major concerns of today's society. In recent years, the European Union has been changing legislation and implementing policies aimed at promoting its investment and encouraging its use in order to reduce the emission of greenhouse gases [1]. This leads to the emergence of a new paradigm in the energy sector, where there is a strong growth of microgeneration, which injects greater complexity into the Energy Markets (EM). Now, the entities that usually were consumers can also be producers, selling surplus energy to the network. As a result, new challenges arise, particularly in the production, distribution, storage and consumption of energy.

By studying data collected from the network, it is possible to formulate strategies that make the system more sustainable, reliable and efficient, preventing waste and minimizing resources [2]. The use of simulators that use this information as a basis is an essential tool for decision support. However, the high complexity characteristic of the sector becomes a challenge [3] because there are several dimensions that influence the behavior of EM, and most of these tools are focused on a specific area of the problem.

It is in this context that the Tools Control Center (TOOCC) emerges, a tool that allows interoperability between heterogeneous multi-agent systems [4], in order to function as a single system. Thus, the various systems, focused on different problems, can work together to study energy systems, allowing the simulation of scenarios with a high degree of complexity.

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 641794 (project DREAM-GO), from the CONTEST project - SAICT-POL/23575/2016; and from FEDER Funds through COMPETE program and from National Funds through FCT under the project UID/EEA/00760/2013.

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 353–356, 2018.
https://doi.org/10.1007/978-3-319-94580-4_37

2 Main Purpose

TOOCC is a multi-agent tool designed to allow the strategic communication of heterogeneous energy systems. The combination of their individual capacities creates a super system, providing results for more complete and complex scenarios, allowing to carry out more realistic studies on the sector. However, it is also possible to execute these systems/algorithms individually. Thus, TOOCC acts as a central entity, responsible for the setup, execution and analysis of different scenarios, which can use one or more systems, depending on the user objective.

To perform the simulation, TOOCC creates an agent for each scenario to execute, which is responsible for establishing communication with the required systems. The communication is made through ontologies, allowing the use of the same vocabulary in their interaction. In this way, it is guaranteed that the systems are able to understand each other and act in the way that is expected.

Currently, TOOCC is integrated with several energy systems, namely the Intelligence and Decision Support multi-agent system (IDeS), which executes the different demand response (DR) [5], optimization, scheduling, forecasting, and decision support algorithms; Multi-Agent System for Competitive Electricity Markets (MASCEM) [6], that runs electricity market simulations; Adaptive Decision Support for Electricity Market Negotiation (AiD-EM) [7], which provides intelligent support for player's decisions in electricity market negotiations; Network Manager (NM) [8], that enables the energy management for a grid (Smart/Micro); Facility Manager (FM) [9], that manages facilities' energy resources; and Programmable Logic Controller Multi-Agent System (PLCMAS). However, the use of ontologies allows other external systems to easily communicate and interact with those presented here. Figure 1 intends to present an overview of TOOCC execution process.

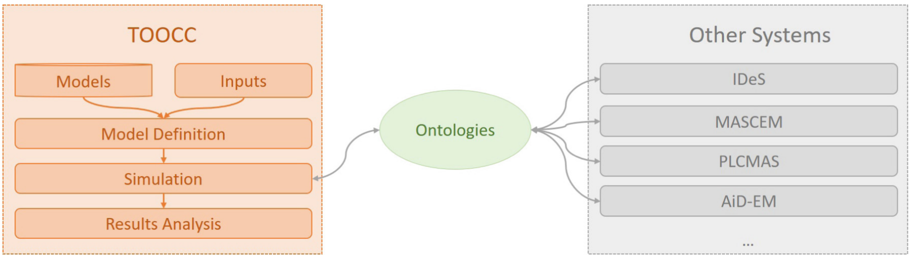


Fig. 1. TOOCC overview

In addition, TOOCC has a mechanism for scheduling agents, guaranteeing that they will be (re)allocated to a machine that has the processing capacity, as well as the software needed to perform its task.

3 Demonstration

Due to the dynamism of the configuration of a scenario, TOOCC has a graphical interface that allows the user to configure it, in a process consisting of three phases: modeling, simulation and analysis. The user has a left menu which allows to user go back and make changes.

In the modeling phase, the scenario is fed by stored models and input data. These models include the definition of distributed network components (storage units, loads and electric vehicles), DR programs, energy tariffs, consumer definition, among others. In turn, the input data includes the parameterization required for the correct functioning of the systems/algorithms. Figure 2 shows the TOOCC panel that allows choosing, defining and redefining different models. In this case, the concerned model refers to energy Tariffs. In this example, a specific tariff from Belgium has been chosen, so that it can be changed and adapted according to the needs and objectives of the simulation scenario to be performed.

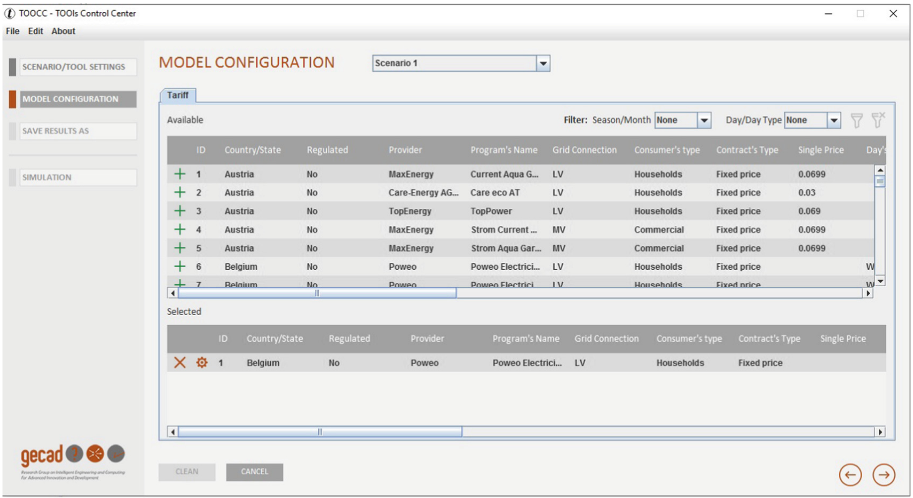


Fig. 2. Graphical interface of TOOCC – Tariffs Definition

In the simulation phase, the agent responsible for the execution of the scenario will communicate with the necessary systems. For this, it uses ontologies designed for this purpose, which are available in [10]. During execution, agents may need to request a machine change in order to continue the simulation.

Finally, the last phase allows the user to analyze the results obtained from the execution, through graphs and drawn tables. These charts and tables can be saved for future use.

4 Conclusion

The growth in the use of renewable energy sources is increasing the complexity of EM. In this way, it is essential that its players can use mechanisms to support decision making, in order to deal with the unpredictability of the sector. There are several simulators that allow the study of EM, however, in that they act to respond to a specific problem.

In order to study the impact of all variables in EM, the TOOCC tool is proposed for the interoperability of heterogeneous energy systems, in order to allow the formulation of more complete and complex scenarios through the use of ontologies. In addition, this tool has a set of characteristics that gives it a great dynamism, because it allows the definition of the scenarios, and the configuration of models, which introduces the specification of simulation scenarios with very distinct natures and characteristics.

References

1. European Commission: 2030 framework for climate and energy policies. http://ec.europa.eu/clima/policies/2030/index%7B_%7Den.htm (2014)
2. Yang, S.X., Wang, Y.: Applying support vector machine method to forecast electricity consumption. In: International Conference on Computational Intelligence and Security, vol. 1, pp. 929–932, November 2006
3. Conejo, A., Carrion, M., Morales, J.: Decision Making Under Uncertainty in Electricity Markets. Springer, Heidelberg (2010). https://doi.org/10.1007/978-1-4419-7421-1
4. Santos, G., Pinto, T., Vale, Z., Praça, I., Morais, H.: Enabling communications in heterogeneous multi-agent systems: electricity markets ontology. ADCAIJ: Adv. Distrib. Comput. Artif. Intell. J. **5**(2) (2016)
5. Faria, P., Vale, Z., Baptista, J.: Demand response programs design and use considering intensive penetration of distributed generation. Energies **8**(6), 6230–6246 (2015). http://www.mdpi.com/1996-1073/8/6/6230
6. Santos, G., Pinto, T., Praça, I., Vale, Z.: MASCEM: optimizing the performance of a multi-agent system. Energy **111**(C), 513–524 (2016). http://econpapers.repec.org/RePEc:eee:energy:v:111:y:2016:i:c:p:513-524
7. Pinto, T., Morais, H., Sousa, T.M., Sousa, T., Vale, Z., Praca, I., Faia, R., Pires, E.J.S.: Adaptive Portfolio optimization for multiple electricity markets participation. IEEE Trans. Neural Netw. Learn. Syst. **27**, 1720–1733 (2016)
8. Silva, M., Morais, H., Sousa, T., Faria, P., Vale, Z.: Time-horizont distributed energy resources scheduling considering the integration of real-time pricing demand response. In: IEEE Eindhoven PowerTech. IEEE, pp. 1–6 (2015)
9. Fernandes, F., Morais, H., Vale, Z., Ramos, C.: Dynamic load management in a smart home to participate in demand response events. Energy Build. **82**, 592–606 (2014)
10. GECAD: GECAD's Inteligent Energy Systems Ontologies. http://www.gecad.isep.ipp.pt/ontologies/ies/ (2017)

MATISSE 3.0: A Large-Scale Multi-agent Simulation System for Intelligent Transportation Systems

B. Torabi[1(✉)], M. Al-Zinati[2], and R. Z. Wenkstern[1(✉)]

[1] University of Texas at Dallas, Richardson, TX, USA
{behnam.torabi,rymw}@utdallas.edu
[2] Jordan University of Science and Technology, Irbid, Jordan
mhzinati@just.edu.jo

Abstract. In this demo we present MATISSE 3.0, a microscopic simulator for agent-based Intelligent Transportation Systems (ITS). MATISSE provides abstract classes for the definition of vehicle and intersection-controller agents as well as components for the concurrent 2D and 3D visualizations of the simulation. The control GUI allows the user to change various agent properties at run time. The OSM converter imports entire traffic networks from Open Street Map. We illustrate the use of MATISSE through the development of a simulation for the City of Richardson, Texas.

Keywords: Agent-based simulation
Intelligent Transportation Systems · Traffic simulation

1 Introduction

Intelligent Transportation Systems (ITS) offer significant opportunities to improve traffic safety, relieve traffic congestion and reduce air pollution. While in-vehicle technologies and infrastructure systems have been developed, the full potential of ITS can only be realized when advanced sensing devices, autonomous smart systems, and advanced communication mechanisms (i.e., V2V, V2I, I2I) are available. As various cities around the world have started implementing ITS features, one main concern remains the assessment of the large-scale deployment of new traffic technologies.

Although several microscopic traffic simulators are available [3,6,7], most were not developed around the agent concept and require important development effort to be "agentified" (e.g., [7]). Only a very few are agent-centric (e.g., [4,5]). Unfortunately, these simulators do not offer core agent simulation mechanisms such as sensing, diverse communication types, or the capability to model collisions.

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 357–360, 2018.
https://doi.org/10.1007/978-3-319-94580-4_38

The simulation of ITS calls for the development simulation systems where sensing, autonomy, communication and decentralization are central. In this paper, we present MATISSE 3.0, a large-scale, microscopic agent-based simulation system for ITS. In MATISSE, vehicles and intersection controllers are modelled as virtual agents which perceive their surroundings through sensors, and continuously interact with one another.

2 High-Level Architecture

MATISSE's architecture consists of three building blocks (see Fig. 1) [1,2]. The simulator's main constituent is the *Simulation System* which includes three subsystems: (1) The *Agent System* creates and manages simulated vehicles and intersection controllers as well as zone managers. The various agent types communicate through the Agent-to-Agent Message Transport Service; (2) The *Environment System* creates and manages the traffic environment in which the agents are situated; and (3) the *Simulation Microkernel* manages the simulation workflow.

The *Message Transport Service* provides a configurable messaging infrastructure that allows MATISSE's building blocks to exchange information with each other.

The *Control and Visualization System* renders 2D and 3D representations of the simulation and provides real-time interaction mechanisms.

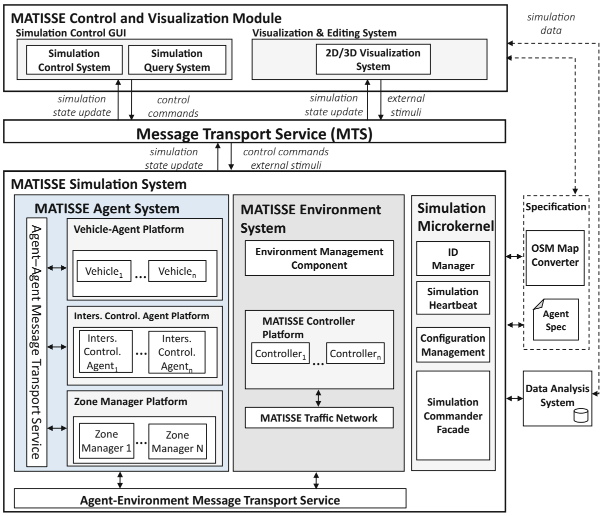


Fig. 1. MATISSE's high-level architecture

3 MATISSE's Features

A demo illustrating MATISSE's features is available at: http://www.utdmavs.org/matisse3.

To create a virtual agent, the modeler can either create a concrete class extending from abstract classes provided by MATISSE, or instantiate one of the pre-defined concrete classes.

To execute the simulation, we first start the simulation service then we use the control panel to control the simulation and start its supporting applications.

The user imports a road network from Open Street Map. A network area can be selected either from a file system or an online map viewer. Once the selected network is imported, MATISSE uses advanced algorithms to automatically generate missing information such as unknown road types or traffic light locations. The final network is then imported into the simulation.

At the start of the simulation, the user defines the total number of vehicles to be run. Vehicles enter and exit the simulation from entry and exit points. The user can let MATISSE generate an initial normal distribution or define their own vehicle distribution for preferred entry and exit points.

Virtual autonomous vehicles are equipped with sensors. They perceive and communicate with agents located within their sensor range, called *circle-of-influence*.

Communications between autonomous vehicles and intersection controllers can be visualized during the simulation.

With MATISSE, we can modify agent properties such as circle-of-influence and speed and display these properties at run-time. Also, we can track agents by their IDs and change properties of a group of agents.

For non-autonomous vehicles, a driver-agent is defined. The virtual driver's perception allows the computation of the driver's level of distraction.

During the simulation, the user can modify the driver's level of distraction. This may dynamically introduce unexpected accidents and unpredicted traffic behavior.

4 Case Study

In this Section, we briefly discuss the simulation of an intelligent traffic signal system for the City of Richardson, in Texas. The city consists of 1365 road segments and 128 signalized intersections. The city's intersection controllers are managed by a central traffic management center.

The experiments were run on a multicore PC (Intel Core i7 X980CPU (3.33 GHz), 6.00 GB, 64-bit Windows 7). Using traffic data provided by the City of Richardson, we executed two simulation scenarios six times for 86,400 simulation cycles each. The demo shows the two simulation scenarios for 3,000 human-driver agents. In the first scenario, we simulate the current centralized traffic infrastructure with virtual inductive loop sensors, and conventional intersection controllers operating in fully-actuated mode. In the second scenario, we simulate a decentralized agent-based infrastructure, where intersection-controller are managed by agents which autonomously communicate and coordinate actions to alleviate congestion.

The simulation results show that the agent-based traffic-control system outperforms the conventional fully-actuated system.

5 Conclusion

MATISSE is an agent-based traffic simulation system. In MATISSE, autonomous vehicles, vehicle drivers and intersection controllers are modeled as virtual agents. MATISSE offers a variety of features to facilitate the simulation of ITS: predefined abstract classes for various agent types; simulated sensor-based perception mechanisms for agents; simulated I2I, V2V, V2I communications; modification of agent properties at run-time without interrupting the simulation; the emergence of unexpected accidents at run-time; and finally, the automatic conversion of Open Street Map road networks.

As more features are developed, MATISSE will provide a unique fully agent-based framework for the simulation of a variety of real-world traffic scenarios.

References

1. Al-Zinati, M.: Self-organization in multi-agent based simulation systems and its application to intelligent transportation systems. Ph.D. thesis, University of Texas at Dallas, April 2015
2. Al-Zinati, M., Wenkstern, R.: Matisse 2.0: a large-scale multi-agent simulation system for agent-based ITS. In: Proceedings of the 2015 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2015, pp. 328–335, December 2015
3. Intelligent Transportation System Lab at MIT: MITSIMLab. https://its.mit.edu/software/mitsimlab. Accessed Mar 2018
4. MATSim Community: MatSim. https://www.matsim.org. Accessed Mar 2018
5. da Silva, B.C., Bazzan, A.L.C., Andriotti, G.K., Lopes, F., de Oliveira, D.: ITSUMO: an intelligent transportation system for urban mobility. In: Böhme, T., Larios Rosillo, V.M., Unger, H., Unger, H. (eds.) IICS 2004. LNCS, vol. 3473, pp. 224–235. Springer, Heidelberg (2006). https://doi.org/10.1007/11553762_22
6. Integrated Systems and Systems Design: VISSIM. http://www.issd.com.tr/en/22992/PTV-VISSIM-Traffic-Simulation-Software. Accessed Mar 2018
7. Institute of Transportation Systems at the German Aerospage Center: SUMO. http://sumo.dlr.de/index.html. Accessed Mar 2018

The SAIL Framework for Implementing Human-Machine Teaming Concepts

Bob van der Vecht[1(✉)], Jurriaan van Diggelen[1], Marieke Peeters[1], Wessel van Staal[2], and Jasper van der Waa[1]

[1] TNO, The Hague, The Netherlands
{bob.vandervecht, jurriaan.vandiggelen, marieke.peeters, jasper.vanderwaa}@tno.nl
[2] Eagle Science, Amsterdam, The Netherlands
W.vanStaal@Eaglescience.nl

Abstract. Human-machine teaming (HMT) is a promising paradigm to approach situations in which humans and autonomous systems must closely collaborate. This paper describes SAIL, a software framework for implementing HMT-concepts. The approach of SAIL is to integrate existing autonomous systems in a framework, that serves as a social layer between autonomous systems and human team members. The social layer contains reusable modules to provide social capabilities enabling teamwork. The players and modules in the framework communicate via a human-readable communication language that has been developed for HMT concepts. We demonstrate the SAIL framework for a proof of concept task where a human operator is teaming up with a swarm of drones.

Keywords: Human-machine teaming · Social artificial intelligence
Software framework

1 Introduction

This paper describes the software framework SAIL, which stands for Social Artificial Intelligence Layer. SAIL can be used to implement and experiment with Human-Machine Teaming (HMT) concepts. The software layer aims to convert humans and autonomous systems into effective teamplayers.

Current autonomous systems are typically designed to meet the objectives of a specific problem in a specific domain, and are lacking social capabilities required for teamwork. Using SAIL, an HMT is developed as an additional layer of social intelligence which complements the autonomous capabilities of the system. The social capabilities to interact and collaborate with the autonomous systems are made available by connecting the systems to the SAIL framework. For a broader description the use of SAIL as experimentation method for developing HMT concepts, we refer to [1].

In this paper we describe the workings of SAIL using a running example of a human operator who teams up with a swarm of autonomous unmanned air vehicles (UAVs) to execute a force protection task.

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 361–365, 2018.
https://doi.org/10.1007/978-3-319-94580-4_39

2 Use Case: UAVs for Force Protection

To illustrate the use of the SAIL framework, we have implemented a simulated futuristic defense task, where a human operator has the task to protect a ship from incoming hostile drones. He has a number of UAVs available.

The UAVs are able to autonomously execute the force protection task. They have an individual behavior model, which is a finite state machine with three states for surveillance, identification and engagement. Figure 1 shows the UAV behavior model. In the simulated task environment this model has been implemented using the java-based agent framework Mason [2].

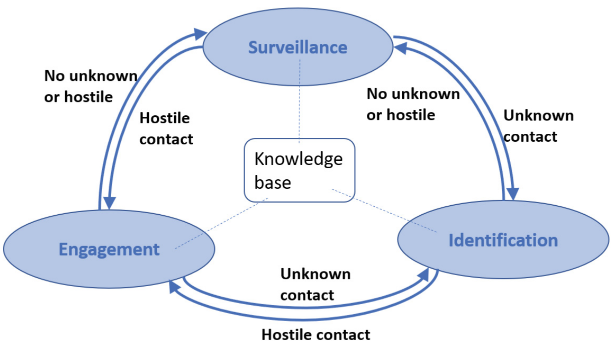


Fig. 1. The behavior model of UAVs is a finite state machine

In the surveillance state a UAV shows a simple surveillance behavior within a predefined area, while maintaining a distance of other UAVs. If an unknown contact is detected, the UAV has to identify whether the contact is neutral or hostile. Therefore it moves closer until the contact is within the identification range. If the contact is identified as hostile, the UAV will start an engagement, which will destroy the hostile contact. The UAVs have no capabilities to exchange information, and therefore base their actions on local knowledge, which lead to suboptimal coordination. The goal is to set up an HMT in which the human operator is able to optimize the performance by coordinating the UAV behavior.

3 Setting up an HMT in SAIL

SAIL functions as a middleware between autonomous software components. It organizes the communication between the components and maintains a registry of active components. The framework is based on Akka [3], a Java toolkit for distributed applications. Components can exchange text messages in JSON format. Communication between modules runs via sockets. The components that are connected to the framework can be implemented in different programming languages, as long as the basis connection interface is provided.

The SAIL method, described in [1], facilitates development of reusable HMT modules that fulfill social function that are used for team work. These HMT modules are linked into the SAIL framework. Through the SAIL GUI, a human-agent team can be configured by defining the active actors, modules and the lines of communication between them.

For the force protection task, an HMT configuration is set up as shown in Fig. 2. The components in this HMT set up are the user and his/her main interface, the UAVs as the autonomous systems and five generic HMT modules, implemented as python scripts that are wrapped in SAIL, of which some are interconnected as they extend each other's capabilities:

(1) Situational Awareness Module; tracks the position, action and knowledge of each UAV.
(2) Asset Grouping Module; allows the user to group assets based on any criteria.
(3) Work Agreements Module; allows the user to use predefined work agreements for any UAV or group of UAVs due to the Asset Grouping Module.
(4) User Identification Module; an example of a more specific HMT module that allows the user to change specific information (identification) in the UAVs.
(5) External Messages Module; listens to any message send from either the UAVs or from others outside the team, and forwards it to the user interface.

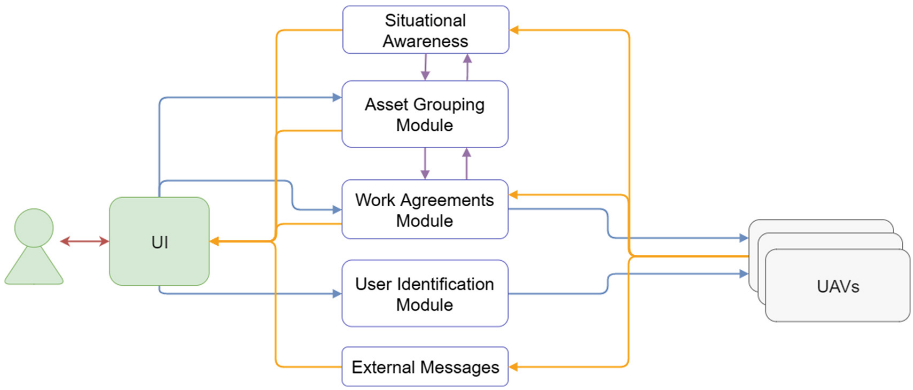


Fig. 2. Example configuration of SAIL. Graphical representation.

4 Communication with HATCL

The actors and modules in the SAIL framework need to communicate and understand each other. HATCL (Human-Agent-Teaming Communication Language, [1]) serves as a communication language specifically designed for human machine teaming. The initial version of HATCL that specifies the message format and domain ontology in plain old java (pojo) classes. Messages in HATCL contain a performative specifying the intention of a message, such as inform, request, confirm and also permissions, prohibitions and obligations in a directive setting.

It is required that the actors are designed to process HATCL messages properly. Therefore, an wrapper for the UAVs is defined that make the UAVs interpret and produce HATCL messages. The wrapper serves as interface between SAIL and the UAVs, and implements semantic anchors, that define how HATCL constructs map to the internal UAV behavior model.

By creating semantic anchors for the UAVs, they can interpret messages from the HMT modules. The following HATCL performatives have been created for the UAVs:

- *Prohibition*: a prohibition of an action blocks transitions to the corresponding state
- *Permission*: a permission of an action allows transitions to the corresponding state
- *Obligation*: an obligation enforces the transition to the corresponding state
- *Inform*: an inform message with the identification of a contact overrules the internal identification of the UAV

5 Prototype

The above description results in a working HMT for the force protection task. The combination of the modules allows new team coordination possibilities between the human user and the swarm of UAVs. For example, a user is able to define a group *North* or *South* and set up a working agreement that this group is prohibited to engage. Although the UAVs themselves have no notion of the concept of groups or geographic areas or work agreements, the modules provide an interface for the user to coordinate with the UAVs using these concepts. Figure 3 shows the user interface for the operator.

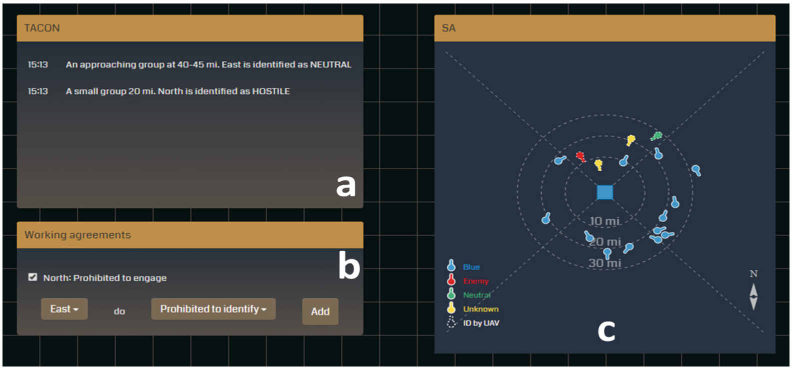


Fig. 3. The UI, via which the operator coordinates with the UAVs, with modules for information messages (a), working agreements (b) and situation awareness (c).

References

1. Van der Vecht, B., Van Diggelen, J., Peeters, M., Barnhoorn, J., Van der Waa, J.: SAIL: a social artificial intelligence layer for human-machine teaming. In: PAAMS 2018 (2018)
2. Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., Balan, G.: Mason: a multiagent simulation environment. Simulation **81**(7), 517–527 (2005)
3. Akka toolkit. https://akka.io/

Check for updates

A Novel Web Services Infrastructure Leveraging Agent Oriented Middleware Environment for Realizing Agent Oriented Information System Models

D. Venkatesan(✉) and S. Sridhar

Anna University Chennai, Chennai, India
d.venkatesan@aubit.edu.in

Abstract. Recent industry surveys have shown complete disenchantment with unified modeling language (UML) as the choice of system modeling and development. UML is a standard promoted by industry consortium Object Management Group (OMG) which also keeps it at the center of several other modeling standards (such as SysML), has little room for retreat. This situation lead to a vacuum in the model based enterprise software system development. This work demonstrates a proposed alternative to UML based system development using ontologically natural agent oriented modeling constructs and show cases a realization environment of the agent oriented (information) system models using a middleware technology called as System Incorporating Versatile Autonomous Nodes (SIVAN). SIVAN sustain an ecosystem of agents, organizations and societies. SIVAN is implemented in C#/.Net/WCF technology and makes use of several web services (WS) standards/technologies for communication, eventing, serialization, and security.

Keywords: Speech acts · Middleware based web service orchestration
Application integration · Loose coupling · Decoupled orchestration
Provider-consumer · Case study · WS infrastructure

1 Introduction

Gorschek et al. (2014) and Marian (2014) state due to several structural and methodical deficiency, UML could not meet goals of enterprise modeling. Venkatesan (2018) suggests an agent metaphor based information system modelling. Central to that proposal involve framing an ontologically natural agent oriented information system modeling language namely AOML. AOML utilizes ontological constructs that are natural in business domain models and consists of agents (A), organization (O), society (S), speech acts (SA), conversation (C), system property (AOSSACSP) to encode system requirements and permit system wide development (by stepwise refinement) without any syntactic discontinuity between model diagrams and model diagram kinds. Central to the realization of this agent oriented environment is the agent container middleware engine that to best extent possible should leverage standards and infrastructure provided by developments in software technology such as web services

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 366–370, 2018.
https://doi.org/10.1007/978-3-319-94580-4_40

standard (Venkatesan and Sirshar 2017). Venkatesan (2018) presently utilizes C#/. Net/WCF environment to realize SIVAN. As SIVAN may be of interest to the agent community, this work presents its salient aspects and provides a demonstration of its capability through an actual real-life working prototype case study. Before getting into the details of SIVAN, this work provides an overview of AOML to enable readers appreciate and anticipate capabilities of SIVAN. Complete detail of this system and its theory is provided in Venkatesan (2018), along with a scholastic treatment with respect to the state of the art in agent technology.

2 Agent Oriented Modeling Language and Its Enactment/Realization Environment 'SIVAN'

The AOML presented in Venkatesan (2018) revolves around white-box system requirements modeling (using AOSSACSP) of free text business information system user requirements (whereas UML use-cases are black box models!). AOML is an ontologically natural information system model in the sense of Bungee-Wand-Weber criterion for naturalness of IS models. The following descriptions are to be read in the context of AOML capabilities. Speech acts are realized using FIPA like messages while "conversations are nothing but business functionality model (akin to "activity diagram in UML or BPMN scenario). Organizations (following the definition of Dignum 2009) are captured as links to contained agents/types with ability to pump SA/C messages to all contained types just like a window-class in WIN32 (of MS-Windows API) process message queue among its embedded HWND classes. Societies are captured as agents of same type but belonging to different concrete instances that permit choice based competitive service to client agents (For example, a purchaser agent can make use of any one of payment service agent (PSA) in PSA society. Agent abstraction consists of any active entity that has behavior (in which case will have some SA or C) or plain properties alone (in the case of non-mutable static entity). 'Agent" is the sole and only type in the entire system model facilitating specialized static (compiler based) or dynamic (middleware based) enforcement of (system wide) functional or non-functional policies/properties.

For enacting the agent based business models, there is a need for agent container deployment engine and a runtime environment. Unlike JADE that contains/hosts instances of Java classes as agents, SIVAN agents are more of an independent C# applications running autonomously and connecting with SIVAN middleware through WCF messages. AOML supports two new architectural unit primitives namely *'migration'* and *'loose coupling'* (Venkatesan and Sridhar 2017, 2018) using which system model can be weaved.

2.1 SIVAN Environment

In SIVAN ecosystem, any compute node capable of running agents must run an instance of SIVAN service. SIVAN service can be run in global core service mode (that maintain addresses of all other core servers) and replicates agent name space on all services to enable SA routing) or local service mode (that maintains bidirectional link to a single global core SIVAN service (for routing message requests) but also a local agent name server that maintain registry of local agent services available and their instances. Local instance of SIVAN server permit fast computation avoiding costly message based network invocation. Present version of SIVAN makes use of C#/. Net/WCF programming environment (SIVAN 2018) and runs on MS-Windows configuration files to link up with core servers, and core servers themselves boot-strap using configuration files (Fig. 1).

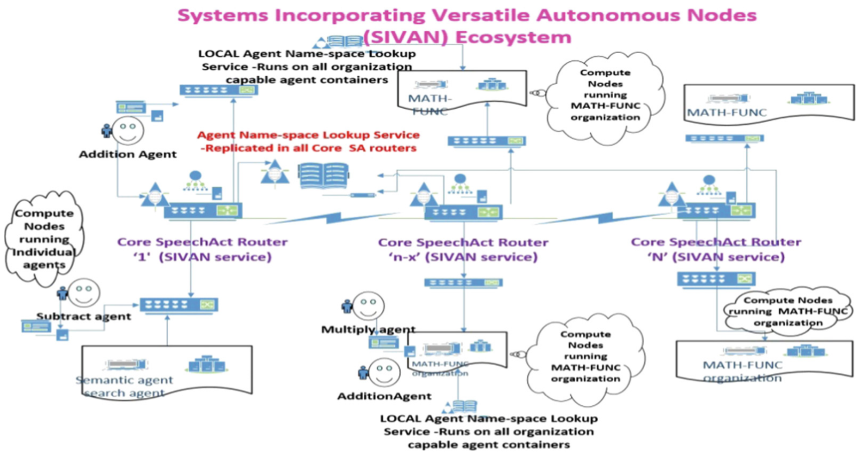


Fig. 1. Block diagram of SIVAN agent ecosystem

2.2 Demonstration of MATHTUTOR Case Study

It is easiest to understand SIVAN environment through a practical and intuitive use case namely MATHTUTOR (Chap. 3 of Venkatesan 2018) while more complex aspects of AOML model types such as organization and society warrant a more complex use case such as MEDPLAN (Chap. 6 of Venkatesan, 2018). MATHTUTOR (Fig. 2) use case involve a teacher administering a basic mathematics skill test for his students in their ability to solve addition, subtraction and multiplication (using traditional client-server WS orchestration paradigm (Fig. 2A) and agent-oriented computing paradigm (Fig. 2B)). MATHTUTOR case involve a client (i.e. consumer) (CLNT) supplying a packet of data consisting of rows of two numbers and a place holder for storing result and computing the result using client server technology (Fig. 2A). Figure 2B illustrates sending the agent itself around three compute nodes CN1, CN2 and CN3 for carrying out node specific computation. CN1 hosts "addition" web service, CN2 hosts "subtraction" web service and "CN3" hosts "multiplication" web service. For *each* student (simulated locally in CNs as problem solvers) that undertake this test, *an instance of MATHTUTOR agent gets invoked* and it administers add, subtract and multiply tests. The purpose of this use case is to illustrate superiority of the agent oriented modeling.

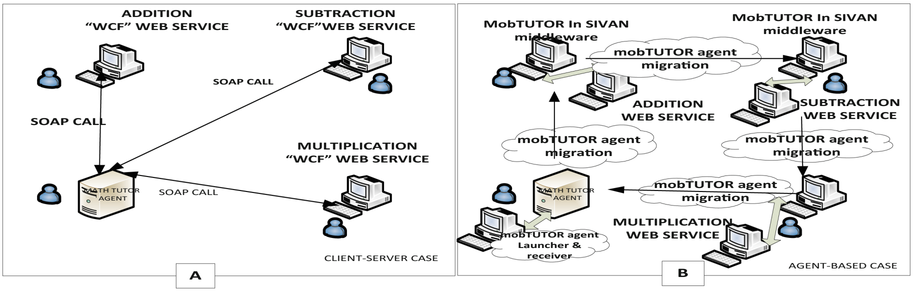


Fig. 2. MATHTUTOR case involving compute nodes CN1, CN2, CN3 and CLNT CN

The logic of MATHTUTOR agent based application design case study encoded using agent script is given below as pseudo-code in Fig. 3.

```
Precondition: On CLNT CN math-tutor ACIAF script("mobTutor") is invoked.
    1) Receive data packets from user, call it {D}
    2) Migrate to partner link CN1 & register with SIVAN env.
         //CN1  exposes "additionWS"
    3) compute [CA1]:addition test result by invoking   "additionWS" locally using speech act "SA1" & loose cou-
         pling                                    //CN1 hosts additional  "ADD" services (like add1,
         add2 ) to //demonstrate loose coupling in case failure of "additionWS"service.
    4) update ACIAF script state data; {D} = {D} U {CA1}
    5) Migrate to partner link CN2 & register with SIVAN env.
         //CN2  exposes "subtractionWS"
    6) compute [CA2]:subtraction test result by invoking   "subtractionWS" locally using SA2  & loose coupling
         //CN2 hosts additional  "SUBTRACT" services (like sub1, sub2 ) to //demonstrate loose coupling in case
         failure of "subtractionWS"service.
    7) update ACIAF script state data; {D} = {D} U {CA2}
    8) Migrate to partner link CN3 & register with SIVAN env.
         //CN3  exposes "multiplicationWS"
    9) compute [CA3]:subtraction test result by invoking   "multiplicationWS" locally using SA3  & loose coupling
         //CN3 hosts additional  "MULT" services (like mul1, mul2 ) to //demonstrate loose coupling in case fail-
         ure of "multiplicatWS"service.
    10)  update ACIAF script state data; {D} = {D} U {CA3}
    11)  Migrate to partner link CLNT CN  //business case completed; invoker analyze result
```

Fig. 3. Pseudocode of MATHTUTOR script processing logic

3 Conclusion

This work presented the case of agent oriented system modeling and realization using WS standard compliant middleware in the context of failure of UML in industry practice. It demonstrates the merits services technology complaint agent ecosystem.

References

Dignum. V.: Handbook of Research on Multi-Agent Systems. IGI Global (2009)

Gorschek, T., et al.: On the use of software design models in software development practice. J. Syst. Softw. **95**, 176–193 (2014)

Marian, P.: "No shit" or "Oh, shit!": responses to observations on the use of UML in professional practice. Softw. Syst. Model. **13**(4), 1225–1235 (2014)

Okouya, D., et al.: An infrastructure for the design and development of OIS. In: Cossentino, M., et al. (eds.), EMAS 2013, pp 215–234 (2013). https://doi.org/10.1007/978-3-642-45343-4

Venkatesan, D.: A novel agent-based enterprise level system development technology, Ph.D. thesis, AnnaUniv (2018). http://www.openthesis.org/document/view/603477_1.pdf

Venkatesan, D., et al.: A novel programming framework for architecting NGESIS. Inf. Syst. E-Bus Manag. **15**(2), 489–534 (2017). https://doi.org/10.1007/s10257-016-0330-y

A Demonstration of Simulation Modeling for SIoV Recommendations System

Kashif Zia[1], Dinesh Kumar Saini[1], Arshad Muhammad[1](✉), and Alois Ferscha[2]

[1] Faculty of Computing and Information Technology, Sohar University, Sohar, Oman
amuhammad@soharuni.edu.om

[2] Johannes Kepler University Linz, Linz, Austria

Abstract. This paper demonstrates how to model a simulation of Social Internet of Things (SIoV) based recommendation system. An agent-based model of information sharing (for context-based recommendations) of a hypothetical population of smart vehicles is presented. Some important hypotheses are tested under reasonable connectivity and data constraints. We demonstrate how we model our simulation using NetLogo.

Keywords: Internet of Things · Social IoV · Agent-based model Context-awareness

1 Introduction

Vehicular Internet of Things (VIoT) is more than Vehicular Ad hoc NETworks (VANETs) [2] and Vehicular Social Networks (VSN) [1], – whereby, vehicles build and manage their own social network – more appropriately termed as Social Internet of Vehicles (SIoV) [4]. At the application level, SIoV must build and use the network to achieve owners' goals. Towards this, the properties of human social network can be mapped onto buildup and evolution of inter-vehicle social network. This paper provides a demonstration of developing a simulation of an agent-based model of SIoV, integrating modeling components defining IoT connectivity, principles of social network evolution, and users profiling. The models are configured towards an information sharing application for a hypothetical population of smart vehicles using context-based recommendations.

In a grid-based spatial setting, a Point of Interest (PoI) is integrated in users' *plan* using *ID*, *Time* and *Duration*, where *ID* is the identity of the destination, *Time* is the time to reach the destination, and *Duration* is the time to stay at the destination. When a user visits a PoI using its vehicle, if the *quality* of service provided by that PoI is not up to the user's expectation (user *experience*), the PoI is considered as *suspended* for that user. Vehicles connect with each other if they are at the same PoI, thus evolving a social network of vehicles with repeated encounters, consequently resulting in information sharing about the current statuses of PoIs. Following hypothesis are evaluated using simulations:

© Springer International Publishing AG, part of Springer Nature 2018
Y. Demazeau et al. (Eds.): PAAMS 2018, LNAI 10978, pp. 371–374, 2018.
https://doi.org/10.1007/978-3-319-94580-4_41

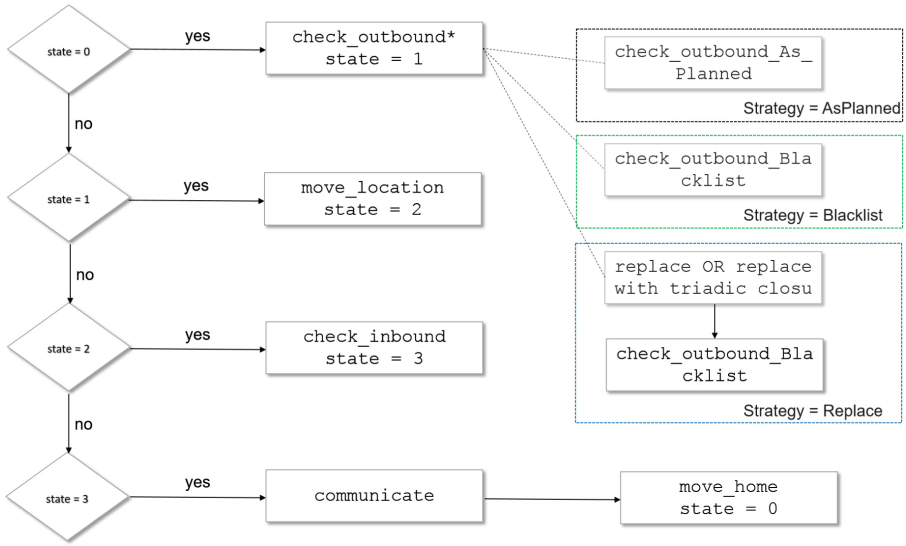


Fig. 1. Vehicle state transitions to be performed at each iteration (= one hour) of the simulation by all the vehicles. If *state* = 0, the vehicle is located at user's home. A vehicle will transit to the next state (*state* = 1) if it is *time* to execute a component of the plan. If *state* = 1, the vehicle moves to the desired location along with transiting to the next state (*state* = 2). If *state* = 2, and the *duration* to stay at the current location is expired, the vehicle will transit to the next state (*state* = 3). If *state* = 3, the vehicle will communicate with its neighbors (in proximity). The vehicles co-located at the same PoI network with each other to form ties. Initially, it is a weak tie. With repeated encounters, a weak tie becomes a strong tie. After communicating, the vehicle will move back to home and resets its state to 0.

- *Hypothesis 1:* Novelty in information is supported by triadic closure, a property of social networking evolution corresponding to more sound connectivity due to repeated encounters [3].
- *Hypothesis 2:* Novel information sharing is directly proportional to how early the information sharing starts.
- *Hypothesis 3:* A fair distribution of resources should be guaranteed as a result of novel information sharing.

2 The Model and Strategies

The basic purpose of the vehicles is to visit PoIs given in the plan at their prescribed time. A list of random PoIs are chosen along with corresponding visit times and visit duration, which are also random. The transition system of vehicles' behavior is divided into five states. Figure 1 presents a graphical view of vehicles' state transitions. The three strategies adopted are: (i) AsPlanned, (ii) Blacklist, and (iii) Replace (without and with triadic closure). In the first strategy, there is no dynamical changes in the plan. The plan is executed as it is. Second strategy works as follows. If the vehicle has to visit a PoI at the current hour and that PoI is not already blacklisted (if blacklisted then do not visit any location), then the quality of current PoI will be evaluated against vehicle's *expectation*. If quality is greater than expectation, the vehicle will transit to state 1, ready to visit the current PoI. Otherwise, the PoI will be suspended in the plan matrix. Third strategy works as follow all those PoIs that have been suspended would be replaced by other PoIs which are not suspended. The information about these PoIs would come from "friends" (having strong ties) of a vehicle. Replacement has two varieties; (i) replacement without triadic closure, and (ii) replacement with triadic closure. The algorithms for strategies blacklist, replacement without triadic closure, and replacement with triadic closure are given in Fig. 2(a), (b) and (c), respectively. A PoI would then be chosen.

```
i ←0
while (i < 3) do
    t ←plan[i,1]
    q ←plan[i,3]
    s ←plan[i,4]
    if (t = current_time&&s = false) then
        if (q > expectation) then
            state ← 1
            select ID of this row as destination
        else
            plan[i, 4] ← true
        end if
    end if
    j ←0
    while j < 3 do
        qq ←plan[j,3]
        if (qq >= expectation) then
            plan[j, 1] ← current_time
            state ← 1
            select ID of this row as destination
            j ←3
        else
            j ←j++
        end if
    end while
end while
```

(a)

```
i ←0
while (i < 3) do
    t ←plan[i,4]
    if (s = false) then
        i ← i++
    else
        j ← 0
        c ← length[vehiclesdata]
        while (j < c) do
            t ← friend j in the list of friends with strong tie
            fplan ← plan of f
            k ← 0
            while (k < 3) do
                ss ← fplan[k, 4]
                if (ss = false) then
                    plan[i] ← fplan[k]
                    k ← 3
                    j ← c
                else
                    k ← k++
                end if
                j ← j++
            end while
        end while
    end if
end while
```

(b)

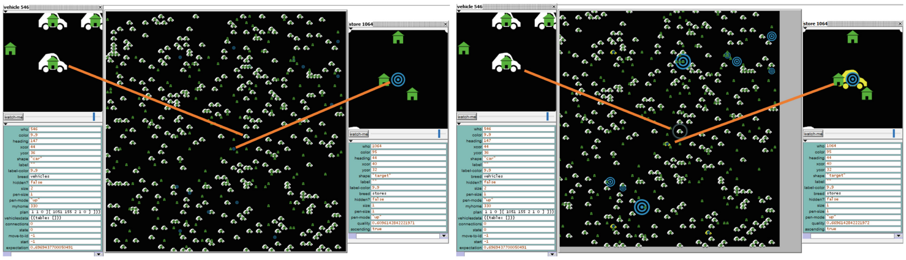
```
nof ← number of friends of this vehicle
if (nof > 1) then
    fv ← random[0 − nof − 1]
    sv ← random[0 − nof − 1]
    if (fv! = sv) then
        if fv has strong tie with this vehicle then
            if sv is not friend of fv then
                make sv friend of fv
                create strong tie between two vehicle
            end if
            if fv is not friend of sv then
                make fv friend of sv
                create strong tie between two
            end if
        end if
    end if
end if
```

(c)

Fig. 2. PoI selection stretegies, (a) blacklist, (b) replacement without triadic closure, and (c) replacement with triadic closure.

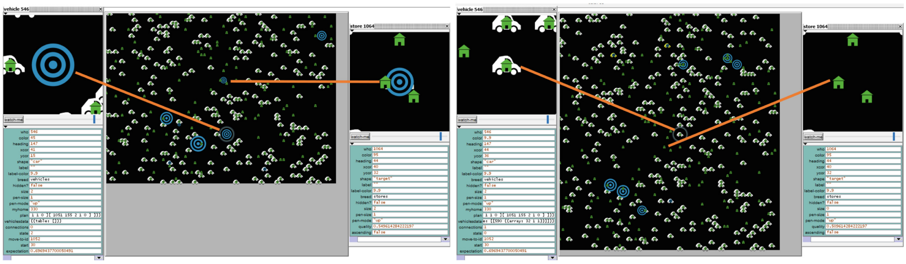
3 Simulation Demonstration

The three strategies adopted (as explained before) are evaluated based on three parameters: (i) quality-index, (ii) connectivity-index, and (iii) PoI-utilization. As discussed in [4], hypotheses 1 and 2 were fully validated by simulation results, whereas, hypothesis 3 was partially validated. In conclusion, closure of social ties and its timing plays an important role in dispersion of novel information. As the network evolves as a result of incremental interactions, recommendations guaranteeing a fair distribution of vehicles across equally good competitors is not possible, particularly at the later stages of interactions (Fig. 3).



(a) At t = 0, showing a vehicle and a store with default parameters settings.

(b) At t = 3, when a vehicle reaches to the store having a specific state shown in yellow color.



(c) At t = 30, when the vehicles moves to a store.

(d) At t = 33, when the vehicle comes back home and store loses its quality and customers.

Fig. 3. Screenshot of simulation demonstrating the dynamics of the changes occurring in a customer's choice of stores.

References

1. Abbani, N., Jomaa, M., Tarhini, T., Artail, H., El-Hajj, W.: Managing social networks in vehicular networks using trust rules. In: 2011 IEEE Symposium on Wireless Technology and Applications (ISWTA), pp. 168–173. IEEE (2011)
2. Cunha, F., Villas, L., Boukerche, A., Maia, G., Viana, A., Mini, R.A., Loureiro, A.A.: Data communication in vanets: protocols, applications and challenges. Ad Hoc Netw. **44**, 90–103 (2016)
3. Easley, D., Kleinberg, J.: Networks, Crowds, and Markets: Reasoning About a Highly Connected World. Cambridge University Press, Cambridge (2010)
4. Zia, K., Muhammad, A., Saini, D.K., Ferscha, A.: Agent-based model of smart social networking-driven recommendations system for internet of vehicles. In: Y. Demazeau et al. (eds.) PAAMS 2018. LNAI, vol. 10978, pp. 275–287. Springer, Cham (2018)

Author Index