Joaquim Filipe
Ana Fred (Eds.)

Communications in Computer and Information Science 358

Agents and Artificial Intelligence

4th International Conference, ICAART 2012
Vilamoura, Portugal, February 2012
Revised Selected Papers

Springer

Communications in Computer and Information Science 358

Joaquim Filipe   Ana Fred (Eds.)

Agents and Artificial Intelligence

4th International Conference, ICAART 2012
Vilamoura, Portugal, February 6-8, 2012
Revised Selected Papers

Springer

Volume Editors

Joaquim Filipe
INSTICC and IPS
Estefanilha, Setúbal, Portugal
E-mail: joaquim.filipe@estsetubal.ips.pt

Ana Fred
IST - Technical University of Lisbon
Lisbon, Portugal
E-mail: afred@lx.it.pt

ISSN 1865-0929 e-ISSN 1865-0937
ISBN 978-3-642-36906-3 e-ISBN 978-3-642-36907-0
DOI 10.1007/978-3-642-36907-0
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013935223

CR Subject Classification (1998): I.2.8-11, I.2.6, H.3.3-5, H.4.1-3, H.2.8, F.1.3

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

Printed on acid-free paper

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

Preface

The present book includes extended and revised versions of a set of selected papers from the 5th International Conference on Agents and Artificial Intelligence (ICAART 2012), held in Vilamoura, Portugal, during February 6–8, 2012, which was organized by the Institute for Systems and Technologies of Information, Control and Communication (INSTICC). ICAART 2012 was held in cooperation with the Portuguese Association for Artificial Intelligence (APPIA), the Spanish Association for Artificial Intelligence (AEPIA), and the Association for the Advancement of Artificial Intelligence (AAAI). INSTICC is member of the Workflow Management Coalition (WfMC), Foundation for Intelligent Physical Agents (FIPA) and the Object Management Group (OMG).

The purpose of the International Conference on Agents and Artificial Intelligence (ICAART) is to bring together researchers, engineers, and practitioners interested in the theory and applications in these areas. The conference was organized in two simultaneous tracks: Artificial Intelligence and Agents, covering both applications and current research work within the area of agents, multi-agent systems and software platforms, distributed problem solving and distributed AI in general, including Web applications, on one hand, and within the area of non-distributed AI, including the more traditional areas such as knowledge representation, planning, learning, scheduling, perception and also not so traditional areas such as reactive AI systems, evolutionary computing and other aspects of Computational Intelligence and many other areas related to intelligent systems, on the other hand.

ICAART 2012 received 292 paper submissions from 52 countries in all continents. In all, 42 papers were published and presented as full papers, 71 papers reflecting work-in-progress or position papers were accepted for short presentation, and another 36 contributions were accepted for poster presentation. Finally, we selected only 28 paper to be included in this book, representing 9.6% of submitted papers.

We would like to highlight that ICAART 2012 also included four plenary keynote lectures, given by internationally distinguished researchers, namely – Anthony G. Cohn (University of Leeds), Frank Dignum (Utrecht University), Luís Paulo Reis (University of Minho/ LIACC), and Wolfgang Wahlster (German Research Center for AI). We would like to express our appreciation to all of them and in particular to those who took the time to contribute with a paper to this book.

We must thank the authors, whose research and development efforts are recorded here. We also thank the keynote speakers for their invaluable contribution and for taking the time to synthesize and prepare their talks. Finally, special thanks to all the members of the INSTICC team, whose collaboration was fundamental for the success of this conference.

December 2012

Joaquim Filipe
Ana Fred

Organization

Conference Chair

Ana Fred	Technical University of Lisbon / IT, Portugal

Program Chair

Joaquim Filipe	Polytechnic Institute of Setúbal / INSTICC, Portugal

Organizing Committee

Patrícia Alves	INSTICC, Portugal
Sérgio Brissos	INSTICC, Portugal
Helder Coelhas	INSTICC, Portugal
Patrícia Duartè	INSTICC, Portugal
Liliana Medina	INSTICC, Portugal
Carla Mota	INSTICC, Portugal
Raquel Pedrosa	INSTICC, Portugal
Vitor Pedrosa	INSTICC, Portugal
Daniel Pereira	INSTICC, Portugal
Cláudia Pinto	INSTICC, Portugal
José Varela	INSTICC, Portugal
Pedro Varela	INSTICC, Portugal

Program Committee

Daghan Acay, Australia
Thomas Ågotnes, Norway
Jose Aguilar, Venezuela
H. Levent Akin, Turkey
Rudolf Albrecht, Austria
Natasha Alechina, UK
Klaus-Dieter Althoff, Germany
Frédéric Amblard, France
Leila Amgoud, France
Andreas Andreou, Cyprus
Tsz-Chiu Au, USA
Snorre Aunet, Norway
Antonio Bahamonde, Spain
Suzanne Barber, USA
Steve Barker, UK
Senén Barro, Spain
Roman Barták, Czech Republic
Teresa M.A. Basile, Italy
Jacob Beal, USA
Orlando Belo, Portugal
Christoph Benzmueller, Germany
Greet Vanden Berghe, Belgium

Carole Bernon, France
Daniel Berrar, Japan
Reinaldo Bianchi, Brazil
Juan A. Botía Blaya, Spain
Andre Pinz Borges, Brazil
Tibor Bosse, The Netherlands
Djamel Bouchaffra, USA
Bruno Bouchard, Canada
Ivan Bratko, Slovenia
Paolo Bresciani, Belgium
Egon L. van den Broek, The Netherlands
Giacomo Cabri, Italy
Silvia Calegari, Italy
Rui Camacho, Portugal
Valérie Camps, France
Jose Jesus Castro-schez, Spain
Patrick De Causmaecker, Belgium
Nicolò Cesa-Bianchi, Italy
Xiaoping Chen, China
Marco Chiarandini, Denmark
Diane Cook, USA
Dan Corkill, USA
Gabriella Cortellessa, Italy
Paulo Cortez, Portugal
Anna Helena Reali Costa, Brazil
Mehdi Dastani, The Netherlands
Darryl N. Davis, UK
Scott A. DeLoach, USA
Enrico Denti, Italy
Julie Dugdale, France
Edmund Durfee, USA
Stefan Edelkamp, Germany
Thomas Eiter, Austria
Fabrício Enembreck, Brazil
Floriana Esposito, Italy
Maria Fasli, UK
Nazim Fatès, France
Stefano Ferilli, Italy
Antonio Fernández-Caballero, Spain
Edilson Ferneda, Brazil
Klaus Fischer, Germany
Roberto Flores, USA
Claude Frasson, Canada
Naoki Fukuta, Japan
Wai-Keung Fung, Canada
Enrico H. Gerding, UK
Joseph Giampapa, USA
Maria Gini, USA
Piotr Gmytrasiewicz, USA
Herman Gomes, Brazil
Madhu Goyal, Australia
Dominic Greenwood, Switzerland
Eric Gregoire, France
Sven Groppe, Germany
Zahia Guessoum, France
Renata Guizzardi, Brazil
Rune Gustavsson, Sweden
Minghua He, UK
Pedro Rangel Henriques, Portugal
Frans Henskens, Australia
Wladyslaw Homenda, Poland
Wei-Chiang Hong, Taiwan
Mark Hoogendoorn, The Netherlands
Ales Horak, Czech Republic
Enda Howley, Ireland
Marc-Philippe Huget, France
Luke Hunsberger, USA
Carlos Iglesias, Spain
Luis Iribarne, Spain
Fuyuki Ishikawa, Japan
François Jacquenet, France
Wojtek Jamroga, Luxembourg
Yanguo Jing, UK
Lalana Kagal, USA
Fernando Koch, Brazil
Martin Kollingbaum, UK
Igor Kotenko, Russian Federation
Tsvika Kuflik, Israel
Luis C. Lamb, Brazil
Anna T. Lawniczak, Canada
Ho-fung Leung, Hong Kong
Luis Jiménez Linares, Spain
Honghai Liu, UK
Shih-Hsi Liu, USA
Stephane Loiseau, France
Gabriel Pereira Lopes, Portugal
Noel Lopes, Portugal
Adolfo Lozano-Tello, Spain
Hongen Lu, Australia

Bernd Ludwig, Germany
José Machado, Portugal
Prabhat K. Mahanti, Canada
Jan Maluszynski, Sweden
Jerusa Marchi, Brazil
Goreti Marreiros, Portugal
Tokuro Matsuo, Japan
Nicola Di Mauro, Italy
Amnon Meisels, Israel
Benito Mendoza, USA
Daniel Merkle, Denmark
Marjan Mernik, Slovenia
Bernd Meyer, Australia
Ali Minai, USA
José Moreira, Portugal
Haralambos Mouratidis, UK
Christian Müller-Schloer, Germany
Radhakrishnan Nagarajan, USA
Tomoharu Nakashima, Japan
Jens Nimis, Germany
Paulo Novais, Portugal
Luis Nunes, Portugal
Andreas Oberweis, Germany
Michel Occello, France
Sancho Oliveira, Portugal
Andrea Omicini, Italy
Stanislaw Osowski, Poland
Djamila Ouelhadj, UK
Nandan Parameswaran, Australia
Andrew Parkes, UK
Krzysztof Patan, Poland
Loris Penserini, Italy
Leif Peterson, USA
Paolo Petta, Austria
Steve Phelps, UK
Eric Platon, Japan
Agostino Poggi, Italy
Ramalingam Ponnusamy, India
Henry Prakken, The Netherlands
Marco Prandini, Italy
Carla Purdy, USA
Mariachiara Puviani, Italy
Franco Raimondi, UK
Marek Reformat, Canada
Martin Rehák, Czech Republic
Alessandro Ricci, Italy
Miguel Rocha, Portugal
Fátima Rodrigues, Portugal
Daniel Rodriguez, Spain
Pilar Rodriguez, Spain
Juha Röning, Finland
Rosaldo Rossetti, Portugal
Ruben Ruiz, Spain
Fariba Sadri, UK
Manuel Filipe Santos, Portugal
Jorge Gomez Sanz, Spain
Jurek Sasiadek, Canada
Ichiro Satoh, Japan
Edson Scalabrin, Brazil
Andrea Schaerf, Italy
Christoph Schommer, Luxembourg
Michael Schumacher, Switzerland
Frank Schweitzer, Switzerland
Murat Sensoy, UK
Peer Olaf Siebers, UK
Flavio S. CorreaDa Silva, Brazil
Ricardo Silveira, Brazil
Olivier Simonin, France
Iryna Skrypnyk, USA
Adam Slowik, Poland
Alexander Smirnov, Russian Federation
Marina V. Sokolova, Spain
Safeeullah Soomro, Pakistan
Armando Sousa, Portugal
Bruno Di Stefano, Canada
Kathleen Steinhofel, UK
Thomas Stützle, Belgium
Toshiharu Sugawara, Japan
Zhaohao Sun, Australia
Boontawee Suntisrivaraporn, Thailand
Pavel Surynek, Czech Republic
Gion K. Svedberg, Sweden
Ryszard Tadeusiewicz, Poland
Nick Taylor, UK
Patrícia Tedesco, Brazil
Vagan Terziyan, Finland
Michael Thielscher, Australia
José Torres, Portugal
Paola Turci, Italy

Anni-Yasmin Turhan, Germany
Franco Turinis, Italy
Paulo Urbano, Portugal
David Uthus, USA
Eloisa Vargiu, Italy
Brent Venable, Italy
Katja Verbeeck, Belgium
Rineke Verbrugge, The Netherlands
Laurent Vercouter, France
Mirko Viroli, Italy
Aurora Vizcaino, Spain
Giuseppe Vizzari, Italy
Dirk Walther, Spain
Yves Wautelet, Belgium
Rosina Weber, USA
Mary-Anne Williams, Australia
Graham Winstanley, UK
Cees Witteveen, The Netherlands
Stefan Woltran, Austria
T.N. Wong, Hong Kong
Franz Wotawa, Austria
Bozena Wozna-Szczesniak, Poland
Feiyu Xu, Germany
Xin-She Yang, UK
Huiyu Zhou, UK
Jean-Daniel Zucker, France

Auxiliary Reviewers

Nicola Capodieci, Italy
João Carvalho, Brazil
Sara Ceschia, Italy
Serhan Danis, Turkey
Richard Dobson, UK
Baris Gokce, Turkey
Benjamin Johnston, Australia
Xudong Luo, UK
Jeroen de Man, The Netherlands
Alessandro Panella, USA
James Parker, USA
Eanes Pereira, Brazil
Carlos Rodriguez-Fernandez, Spain
Konstantin Vikhorev, UK
Marco Wiering, The Netherlands

Invited Speakers

Luís Paulo Reis	University of Minho / LIACC, Portugal
Anthony G. Cohn	University of Leeds, UK
Wolfgang Wahlster	German Research Center for AI, Germany
Frank Dignum	Utrecht University, The Netherlands

Table of Contents

Invited Paper

PART I: Artificial Intelligence

PART II: Agents

Invited Paper

Coordination in Multi-robot Systems: Applications in Robotic Soccer

Luís Paulo Reis[1,2], Fernando Almeida[2,3,4], Luís Mota[2,5], and Nuno Lau[3,6]

[1] EEUM/DSI - Escola de Engenharia da Universidade do Minho,
Departamento de Sistemas de Informação, Universidade do Minho, Guimarães, Portugal
[2] LIACC - Lab. Inteligência Artificial e Ciência de Computadores, Univ. Porto, Porto, Portugal
[3] IEETA - Instituto de Engenharia Eletrónica e Telemática de Aveiro, Aveiro, Portugal
[4] ESTV/IPV - Escola Superior de Tecnologia de Viseu, Inst. Pol. Viseu, Viseu, Portugal
[5] ISCTE-IUL - Instituto Universitário de Lisboa, Lisboa, Portugal
[6] DETI/UA - Dep. Eletrónica, Telecomunicações e Informática, Univ. Aveiro, Aveiro, Portugal
lpreis@dsi.uminho.pt, falmeida@di.estv.ipv.pt,
luis.mota@iscte.pt, nunolau@ua.pt

Abstract. This paper briefly presents the research performed in the context of FC Portugal project concerning coordination methodologies applied to robotic soccer. FC Portugal's research has been integrated in several teams that have participated with considerable success in distinct RoboCup leagues and competitions. The paper includes a brief description of the main RoboCup competitions in which FC Portugal (and associated teams) has participated with focus in the simulation leagues and related challenges. It also presents a complete state of the art concerning coordination methodologies applied to robotic soccer followed by FC Portugal main contributions on this area. The team contributions include methodologies for strategic reasoning, coaching, strategic positioning, dynamic role exchange and flexible setplay definition and execution. These methodologies compose a complete coordination framework that enable a robotic team to play soccer or execute similar tasks.

Keywords: Multi-robot Systems, Multi-agent Systems, Coordination, Robotic Soccer, RoboCup.

1 Introduction

Our main research goal is the development of a formal model for the concept of team strategy for a competition with an opponent team having opposite goals, general enough to be instantiated to various dynamic competitive domains such as distinct RoboCup leagues. Aiming this general objective our research focus is also concerned with developing general decision-making and cooperation models for soccer playing and similar tasks. We have developed several cooperation mechanisms such as Situation Based Strategic Positioning [1, 2, 3, 4] and Dynamic Positioning and Role Exchange Mechanisms [1, 2, 3, 4]. These mechanisms have proven their validity by being adopted by several teams in different leagues, namely by 5DPO [5] and by the 2008's Mid-size champions, CAMBADA [6, 7].

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 3–21, 2013.

Communication languages and protocols, to convey the most relevant information at the right times to players have also been developed. Also, research is focused on intelligent control of players' sensors to achieve maximum coordination and world state accuracy. Online optimization has been used in order to develop a complete set of efficient low-level skills for soccer playing agents and applied in FC Portugal 2d and 3d teams [8-10].

Coaching is an important research topic in RoboCup. We have proposed Coach Unilang – a general language to coach a (robo)soccer team [11]. Our coach conveys strategic information to players, while keeping their individual decision autonomy. We are also working on a coach agent capable to calculate high-level match statistics that may useful for teams to develop opponent modeling approaches [12-14].

FC Portugal is also very concerned with the development of agent evaluation tools like our offline client methodology; WstateMetrics that evaluates the accuracy of world states and Visual debugger used to analyze the reasoning of agents[1, 15]. Evaluation by domain experts using graphical tools is one of the methodologies that we are also pursuing.

We have also developed a framework for high-level setplay definition and execution, applicable to any RoboCup cooperative league and similar domains. The framework is based in a standard, league-independent and flexible language that defines setplays, which may be interpreted and executed at run-time [16-18].

This paper presents an overview of the main coordination methodologies developed for robotic soccer and similar applications with emphasis for the methodologies developed by the FC Portugal RoboCup team. More information about FC Portugal research may be found on the team published papers such as [1-40].

The paper is organized as follows: Section 2 contains a description of the RoboCup international initiative and its interest for researching in coordination of multi-robot teams. Section 3 contains a survey concerning coordination of multi-robot teams. Section 4 presents a brief overview of the research concerning strategic positioning and formations and on the way to perform flexible setplays for coordinating a robo soccer team. Finally some conclusions and future work are described in the last section.

2 RoboCup International Initiative

RoboCup is an international initiative that aims to motivate the research on multi agent systems and intelligent robotics [41]. The RoboCup Federation organizes every year scientific meetings and world robotic competitions in the fields of robotic soccer, robotic search and rescuing, and domestic robots. Some competitions use simulated environments while others use real robots.

In the soccer domain there are the Simulation League 2D and Simulation League 3D competitions, and several real robots soccer competitions, including the Middle-Size League, Small Size League, Standard Platform League (based on Nao humanoid robot from Aldebaran) and Humanoid League. The rescue competitions include the simulated Agent competition and Virtual competition and also a real robot Rescue League. In the field of domestic robots the competition is performed using real robots and it is called RoboCup@Home League.

The soccer competitions, besides having different rules concerning robots and field dimensions differ in autonomy of the robots and robot construction details. While in the Middle-Size and Standard Platform leagues, robots are autonomous and sensors are mounted on the robots, in the Small-Size league a single agent may decide and send commands to every robot of a team. The agent of the Small Size league typically receives information from a camera positioned above the field to detect the robots and ball positions. The Middle-Size league will be explained in more detail, as some of the coordination methodologies developed by FC Portugal have been transferred to the CAMBADA team that competes in this league. In simulated environments there are several competitions that are detailed in the following sections.

2.1 Middle-Size League

In the Middle-Size League two teams of at most 5 real autonomous robots play soccer in a 18x12m field. Robots height is limited to 80cm. Robots horizontal shadow must fit inside a square of 50cm and the weight is limited at 40kg. Robots are completely autonomous, although they are allowed to communicate each other, and all sensors must be mounted on the robots. The environment is color marked i.e. the field is green, the lines and goals are white and the ball used in each competition is announced at least one month before the tournament. Robots must be black except for the markers of each team that must be cyan and magenta.

The mechanical and electrical/electronic solutions found to build the robots play a very important role in the final efficiency to play soccer. Also the vision subsystem is critical for the final performance of the robots. Games are very active and interesting in this league and the top teams exhibit some very interesting coordinated behaviour.

2.2 Simulation 2D League

RoboCup Simulation 2D League is one of the 3 leagues that started the RoboCup official competitions in 1997. In fact a demonstration of the soccer simulator used in this competition had already been performed during the pre-RoboCup 2006. The view of the RoboCup Organizers is to focus this league research at the top-level modules of the soccer robotics problems: the high-level decision and the coordination of teams of, possibly heterogeneous, robots. Over the years the 2D simulator has evolved, including new features and tuning some others, but the core architecture of the simulator is the same as the one used in 1997.

In the Simulation 2D league a simulator, called soccerserver [42], creates a 2D virtual soccer field and the virtual players, modelled as circles. The simulator implements the movement, stamina, kicking and refereeing models of the virtual world. The models in the simulator are a combination of characteristics taken from real robots (ex: differential drive steering) and from humans (ex: stamina model).

Teams must build the software agents that control each of the 11 virtual robots and also a coach agent. The control of the agents is performed by sending commands to the simulator. The main commands are `dash(dPower,dAngle)`, `turn(tAngle)`, `kick(kPower,kAngle)`, `tackle(tAngle)` and `catch(cAngle)` (used only by the goalie). The simulator implements several virtual sensors for each robot and sends the measures of these sensors to the robots periodically. The most important sensor is the vision sensor, but there's also a sense body sensor (that informs the player of its stamina and own speed) and a hearing sensor. Sensory data is in general subject to noise or to some other type of pre-processing that precludes the agents from knowing the exact value of the measures.

The simulation advances in steps of 100ms, meaning that every 100ms the positions and velocities of every player and of the ball are updated by the simulator. Some of the sensory data (like some modes of the vision sensor) is sent to the agents with a different period than that of the simulation update.

Each agent controls only one virtual robot and must coordinate its efforts to make its best contribution for the teams' goals. It is important to note that the knowledge of the various agents about what is happening at a certain moment is not identical, due to noise and restrictions on several sensors (like the angle of vision or the cut-off hearing distance). Also the environment is very dynamic with the opposite team controlling their robots to oppose the teams' goals.

The coach agent receives, from the simulator, the positions of all players in the field and of the ball without noise. However, the coach cannot handle the ball and as severe limitations on its communication with field agents, that make it impossible to control the field robots using the coach information. The coach may have a very significant impact on team performance by giving advice to the field players and using it to perform high-level tasks like tactic analysis and selection or opponent modelling.

Visualization of the games is assured by an independent application that communicates to the simulator to receive the players and ball positions. Fig. 1 show two possible visualizations of the games in the 2D simulation league. All the 3D features of the 3D viewer in Fig. 1 are not modelled in the simulator but inferred by the viewer for better attraction.

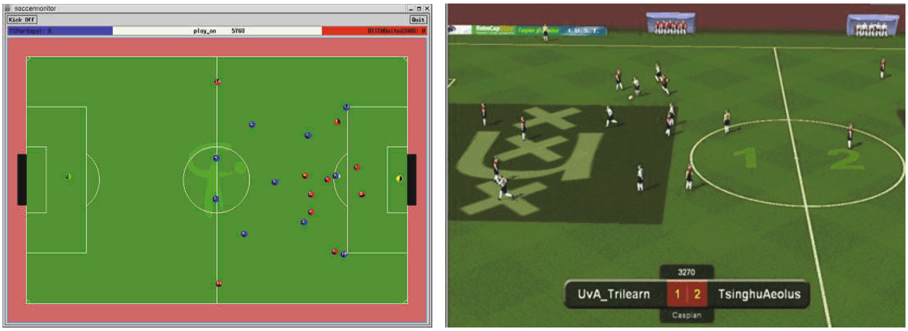


Fig. 1. 2D Simulation League Traditional Viewer (left) and 2D Simulation League Viewer with 3D displaying capabilities ([43]) (right))

2.3 Simulation 3D League

The first version of the Simulation 3D league simulator [44] was made available to the RoboCup community during January 2004. The proposal of the 3D simulator had the following objectives:

- Replace the 2D environment of previous simulator with a 3D environment;
- New, more realistic, physics model;
- Simulation results should not be dependent on available computational power or on the quality of network resources.

Similarly to the 2D simulator, the simulation environment of the RoboCup Simulation 3D League is based on a client-server model. The simulator is the server and agents and visualization tools are the clients. The simulator creates the virtual environment (soccer field, markers, goals, etc.) where agents live, sends sensory information to the agents, receives their actions and applies the virtual physics model in order to resolve positions, collisions and interactions with the ball.

The development of the 3D simulator used available open-source tools extensively. It used the SPADES [45] framework for the management of agent-world communication and synchronization, ODE [46] for the physical model, expat for XML processing, Ruby for scripting language support and boost for several utilities.

Until 2006 the 3D simulation server [47] used the spheres as player model (Fig. 2) and teams had 11 players. A virtual vision sensor sends information about the relative positions of the objects in the world. Replying each sensation an agent sent actions like drive or kick. Driving implied applying a force on the body with a given direction and kicking implied applying a force on the ball radially to the agent. Each sensation was received on every 20 cycles of the server and each cycle took 10 ms.

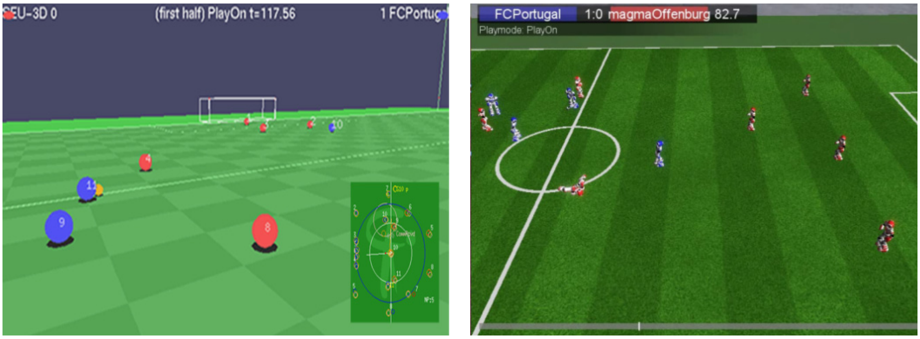


Fig. 2. 3D Simulation league match – sphere model (left); Humanoid robot model (right)

In 2007 the robot model used in the 3D simulator changed from the sphere to a humanoid model. Also the SPADES support was abandoned and a simpler type of timer was developed. The humanoid model used in 2007 was based on the HOAP humanoid robot, then in 2008 the model based on the Nao humanoid robot from Aldebaran has been adopted. The Nao based model has been kept, with very few changes since 2008 until 2012. Due to problems in the simulator, games were played in 2007 between teams of only two players. The number of agents in each team has been increasing over the years and reached 11 players, like in real soccer, in 2009.

3 Coordination in Multi-robot Systems

One of the most important tasks for players is to select and initiate an appropriate (possibly cooperative) behavior (e.g. pass to a teammate, open a pass line) in a given context, using (or not) knowledge from past experiences in order to help their team to win. Good coordination techniques can help achieve this goal, although their success is highly dependent on players individual abilities (low-level skills) to execute adequate competitive decisions. Also, without having a good accuracy on the perception of the world in a real-time, partially observable, stochastic and dynamic environment such as the one in the RoboCup simulated soccer leagues the coordination of player actions would be much more difficult. This section presents an overview of the coordination methodologies applied in RoboCup, in the last years, with emphasis on the simulated 2D league that is, by its characteristics, the best league to research on the coordination area in the context of RoboCup.

3.1 Communication

Communication can be used by players to exchange their intentions and beliefs about the world. The exchange of beliefs helps the players to obtain a more accurate perception of the world, congruent with their teammates. However, not all beliefs have the same importance being, in the soccer domain, the ball location considered the most important [48]. A better knowledge of the world empowers players to execute better suited behaviors.

In several RoboCup leagues a coach agent is available that does not need to rely on communication because he has a global and error-free view of the world. However, typically his communication is highly constrained (low bandwidth and/or significant delays for transmission and reception),making it impossible to rely on his messages to keep an accurate perception of the world state.

Initially, on the simulation 2d league, player-to-player communication allowed the transmission of long messages and players could hear all the messages sent in each cycle. This led to the development of techniques that were not concerned with the selection of which relevant information should be transmitted at any given time, since the bandwidth available was enough to share the most meaningful information about one’s world state with his teammates [49] and communicate useful events/opportunities [1]. The size of messages was shortly reduced to a minimum and the number of simultaneous messages heard in each cycle reduced to one. These new constraints required agents to be capable of cautiously selecting the most pertinent information to send and with their teammates at each instant. Stone et al. [50] addressed the issue of low-bandwidth with the specification of a Locker-Room Agreement (LRA) in which team members have a common understanding of environment cues that trigger predefined strategies. This enabled them to coordinate by minimizing or even eliminating the need for communication. Reis and Lau [1, 2] tried to tackle this issue by measuring the importance of each piece of information through utility metrics based on the current match situation and on the estimated knowledge of teammates. This idea afterward extended in [21] with a Situation Based Communication framework.

Kok et al. [51, 52, 53] proposed the use of a Coordination Graph (CG) as teamwork model for agents to select an optimal joint action and coordinate the execution of their individual actions. This approach requires that each agent communicated his local payoffs for executing a set of individual actions with neighboring agents in order to find a joint action that maximizes the payoff for the team. An extension of this work was proposed to render communication superfluous [51, 53] by assuming that i) players can identify one another; ii) payoff functions are known by all teammates; iii) players can compute the adequacy to fulfill a role for all others; iv) the order of actions is known among all players; and v) in context-specific CGs, all players reachable from a given player can observe the state variables located in his value rules.

Other methods for coordination made use of an explicit communication of belief states [54, 55, 56]. These methods differ on the following criteria: i) communication is assumed to be flawless in [54] but not in [55]; and ii) utility measure of other agents states is estimated based on a sum of heuristic functions represented as potential fields [54] and in [55] it is based on meaningful observed experience.

Stulp et al. [56, 57] proposed the use of temporal prediction models of teammates to enable agents to coordinate for regaining the ball possession. These models are learned offline, using model trees and a neural network, based on the observation of a match with no opponents. These are used to estimate the state of teammates and anticipate the utilities of their intentions in order to adapt to their predicted actions.

Desheng and Kejian [58] proposed a non-communicative approach for coordinating a team of agents using the notions of roles and situation calculus. A role consists on a description of a teammate task and is regarded as a form of intentional cooperation. This approach assumes that an agent is able to compute the adequacy of each agent to fulfill all possible roles based on the situation calculus and consequently figure out the roles that all agents will adopt. Using common knowledge about roles, each individual agent is able to predict the actions of others based on the situation calculus and render communication superfluous.

3.2 Communication Languages

The definition of coaching languages was driven by the need to convey advices from coaches to players during a match and to be able to interact with heterogeneously designed players. The languages Coach Unilang [11], CLang [59], Strategy Formalization Language (SFL) [60] were proposed to structure this communication.

Coach Unilang was proposed on 2001 by Reis and Lau [11] as a generic coaching language enabling high-level coaching of a (robo) soccer team. The language included all features that enable to coach a robo soccer team such as tactics, formations, actions, player types, conditions, regions, periods among others.

CLang [59] was based on the initially proposed Coach Unilang [11] as a simplification in order to be the standard language used by coaches in the RoboCup 2D simulated league and enable to promote a competition focused on simple low-level coaching techniques. It latter evolved to integrate most of the features of Coach Unilang and be a generic coaching language. On this latter version, tactics and behaviors are described using rules which map *directives* (lists of *actions* to execute or avoid) to *conditions* (match situations descriptions). A *condition* is a logical expression based on game variables (e.g. objects positions) whereas an *action* is a low-level skill (e.g. dribble) that a player should be able to execute. When applicable, *conditions* and *actions* can make use of *geometric regions* (e.g. circles) to specify locations in the field such as in Coach Unilang. During a match, these rules are conveyed to players as messages in order to adjust their behaviors.

SFL [60] extends CLang by abstracting its low-level concepts to represent team behavior in a human-readable format that can be easily modifiable in real-time. This language lacks the ability to specify a team's complete behavior with enough detail.

Coach Unilang [11] fixes the inability of CLang to fully specify a team's strategic behavior by enabling the transmission of different types of strategic information (e.g. instructions, statistics, formation, opponent's information) based on real soccer concepts. Players can ignore these messages, interpret them as orders (must be used and will replace existing knowledge) or as advices (can be used with a given trust level).

3.3 Player-to-Player Communication

The standard inter-player communication language CommLang [61], proposed as a complement to CLang in the RoboCup 2D simulated soccer league, promotes player interoperability and eases the creation of mixed-teams of players. This work specifies how to compose and encode different types of information into messages suitable for transmission in the existing limited bandwidth communication channel. However, it does not address i) which types of information should be sent; ii) how often messages should be sent; nor iii) how the information received should be used. Each piece of information is represented by a uniquely identified message type. Current types of messages convey the following information about the sender's beliefs: i) objects positions and velocities; ii) ball ownership; and iii) passing synchronization. A communication message can include one or more types of messages. The use of this language has been shown to improve the confidence and accuracy of a player's world states.

The trend in the robotic soccer domain will be towards little or no communication because it might not always be available, it can be costly and it introduces an overhead and delay that can degrade players performance. The combination of implicit coordination with beliefs exchange has been shown to yield better performance with communication loss than explicit coordination with intentions communication alone [55]. The exchange of beliefs among teammates allows the achievement of more coherent and complete beliefs about the world. These beliefs can be used by a player to predict agents utilities (including his own) and adapt his actions to their predicted intentions to achieve the best (joint) action. As state estimation accuracy reaches an acceptable upper bound it will eventually replace explicit communication for achieving coordination.

3.4 Coaching

In real life soccer, natural hierarchical relations exist among different team members which imply a leadership connotation (e.g. a coach instructs strategy to players). A coach and trainer are privileged agents that can be used to advise players during online games and offline work out (training) situations respectively. Coaching provides a means to monitor and aid the creation of players with *adjustable autonomy* [62] and is an effective method to help a team improve its performance [63].

During a soccer match a coach might have several responsibilities which include i) monitoring teams performance during a match; ii) advising team players to employ appropriate tactics by reusing knowledge from previous games; iii) detect high-level events (e.g. ball possession time) from observing the match observation; iv) classifying opponent strategies (e.g. recognize formations); and v) discovering behavioral patterns of opponent's strengths and weaknesses (e.g. opponent team neglects the right side of their defense when defending). Concerning the advice, players as autonomous agents might decide to follow coach advices or not. On the strategy issue, the sooner the opponent's team strategy is recognized, the sooner the coach can advise his players of the best counter-strategy in order to have a higher impact.

Compared to players, a coach normally has more a priori knowledge, a better view of world and more computational resources available. The communication from coach-to-players during a match can be achieved using structured coaching languages.

Several works have focused on using a coach to improve the performance of a team which included i) changing the team formation based on the score difference, the match remaining time and the ball's path [64]; ii) generating adequate counter strategies based on the modeling of opponent behaviors using classification models based on decision trees [65, 68, 70], neural networks [66], naive Bayes [67] and case-based reasoning [69]; iii) building a marking table that assigns preliminary opponents to each teammate [71]; and iv) recognizing opponents players physical abilities [71].

3.5 Setplays

Setplays can be understood, in a broad sense, as multi-agent plans that need the commitment of several players in order to reach a common goal. Setplays are very common in most sports, e.g., soccer, rugby and handball, which can make one believe that such constructs can also play a useful role in robotic soccer.

The concept of Setplay in RoboCup was first presented in a teamwork and communication strategy for the 2D Simulation league, by Stone and Veloso [50]. These Setplays, however, were quite limited and were meant to be used only in very specific situations, like corner kicks and throw-ins, which are decided by the referee, and are unique for each of these situations.

An interesting approach is presented in Castelpietra et al.[72], where Setplays are represented as transition graphs. These plans, which are formally defined, have a high level of abstraction, and can be applied to different robotic platforms, as it has been the case with Middle-size and four-legged robots. The actual execution of plans and how the robots deal with synchronization issues are unclear topics. In a related research effort [73], Petri Nets have been used to structure the development of a joint team with robots from two distinct institutions.

Kok et al. [52, 53] and Kok and Vlassis [74] in their Coordination Graphs (CG), exploit the dependencies between agents and decomposing a global payoff function into a sum of local terms that are communicated among agents. Nodes in the CG represent agents and its edges define dependencies between them, which have to be coordinated. The continuous aspect of state spaces in robotic soccer discourages the direct application of CGs. To solve this question, roles are allocated to agents to discretize this space and then the CG methods are applied to the derived roles. To simplify the algorithm, it is assumed that only a limited number of near players need to coordinate their actions.

Team Agent Behavior Architecture (TABA) [75] uses hierarchical task decompositions to coordinate the behavior of players in the old four-legged league. Collaboration between players is managed through formations including roles, which describe players' positions. Formations and role choices depend on the team attitude and game state. A CBR system stores a strategy base. A strategy case is a plan designed to achieve a particular goal and includes applicability and termination conditions and a list of formations with roles. Further work by Vega et al. [76] includes a Soccer Strategy Description Symbols (SSDS) graphical notation, an eXtensible Markup Language (XML) behavior language and a control simulator based on Finite State Machines.

4 Strategic Positioning, Formations and Setplays

The selection of a good position during a match is a challenging task for players due to the unpredictability of the environment. However, the likelihood of collaboration in a match is directly related to the adequacy of a player's position. During a match, typically at most one player will carry the ball at each instant. For this reason, players will spend the most time without the ball trying to figure out where to move.

In 1999, Stone [49] proposed a Strategic Positioning using Attraction and Repulsion (SPAR) in which a player maximizes the distance to other players and minimizes the distance to the opponent goal, the active teammate and the ball. Although this approach enabled a player to anticipate the collaborative needs of his teammates but it did not allow the team to assume suitable shapes (e.g. compact for defending) for different situations nor the teammates to have different positional behaviors. Reis et al. [1, 2, 4, 19, 20] proposed a Situation-Based Strategic Positioning (SBSP) method in 2001 to do just that. This method defines team strategy as a set of player roles (describing their behavior) and a set of tactics composed of several formations. Each formation is used in a strategic situation and assigns each player a default spatial positioning and a role. In 2006, Dashti et al. proposed a dynamic positioning based on Voronoi Cells [77] that distributes players across the field making use of attraction vectors to reflect players' tendency towards specific objects based on their roles and the current match situation. Additionally, it does not require the use of home positions nor it limits the number of players per role contrarily to SBSP. In 2008, Akyama et al. proposed a Delaunay Triangulation method [78] inspired by SBSP which divides the soccer field into triangles based on training data and builds a map from a focal point (e.g. ball position) to a desirable positioning for each player. Additionally, this method supports the use of i) constraints to fix topological relations between different sets of training data to compose more flexible formations; ii) unsupervised learning methods to cope with large or noisy datasets; and iii) linear interpolation methods to circumvent unknown inputs. Besides having a good approximation accuracy, is locally adjustable, fast running, scalable and reproducible.

4.1 Defensive Positioning

The main goal of a defending team (without the ball possession) is to stop the opponent's team attack and create conditions to launch their own. In general, defensive behaviors (e.g. marking) involve positioning decisions (e.g. move to intercept the ball). Defensive positioning is an essential aspect of the game, as players without the ball will spend most of their time moving somewhere rather than trying to intercept it.

Initial strategies for choosing an opponent to mark were essentially reactive and consisted on each teammate marking the closest opponent but soon became more elaborate. In 1999, Stone and Veloso [79] proposed that a player who had been assigned the role of team captain as part of a LRA, executes a preset algorithm to decide and communicate to his teammates which opponent they should mark. In 2006, Stolzenburg et al. [71] proposed the use of a centralized and decentralized matching algorithms, to determine which opponent a teammate should mark. These algorithms are executed during non play-on modes and use a distance-based ranking for calculating the matches. A set of teammates and a set of opponents are selected based on their relevance for the current situation and their player types. The centralized approach is executed by a coach which calculates a minimal matching between these sets and informs all players of his results using communication. The decentralized approach is executed by each player who tries to find a partial matching if it exists, but it is susceptible to inconsistencies due to the environment's partial observability which might lead to inaccurate perceptions of the world.

In 2007, Kyrylov and Hou [80] defined collaborative defensive positioning as a multi-criteria assignment problem with the following constraints: i) a set of n defenders are assigned to a set of m attackers; ii) each defender must mark at most one attacker; and iii) each attacker must be marked by no more than one defender. In 2010, they applied the Pareto Optimality principle to improve the usefulness of the assignments by simultaneously minimizing the required time to execute an action and the threat prevented by taking care of an attacker [81]. Threats are considered preemptive over time and are prevented using a heuristic-criterion that considers: i) angular size of own goal from the opponent's location; ii) distance from the opponent's location to own goal; and iii) distance between the ball and the opponent's. This technique achieves good performances while gracefully balance the costs and rewards involved in defensive positioning, but it does not seem to deal adequately with uneven defensive situations such as outnumbered defenders and/or attackers.

Choosing the opponent to mark based only on its proximity might not always be suitable as it disregards relevant information (e.g. teammates nearby) and will lead to poor decisions. Also, the use of a fixed centralized mediator (e.g. coach) to assign opponents to teammates although faster to compute has a negative impact in players autonomy. With the exception of non play-on periods, this approach is not robust enough due to the communication constraints of the robotic soccer domain and because it provides a single point of failure. In 2009, Gabel et al. [82] proposed the use of a NeuroHassle policy to train a neural network with a back-propagation variant of the Resilient Propagation Reinforcement Learning (RPROP-RL) technique in order for a player to learn an aggressive marking behavior which would influence its positioning.

4.2 Offensive Positioning

The selection of positions for players in offensive situations (team owns the ball) typically consists on finding the most suitable position to: i) dribble the ball, for the ball owner player; ii) receive a pass; or iii) score a goal.

In 2008, Kyrylov and Razykov [83] applied the Pareto Optimality principle to the selection of these positions based on the following set of partially conflicting criteria [84] for simultaneous optimization: i) players must preserve formation and open spaces; ii) attackers must be open for a direct pass, keep an open path to the opponent's goal and stay near the opponent's offside line to be able to penetrate the defense; and iii) non-attackers should create chances to launch the attack.

In the same year, Nakanishi et al. [85] proposed a method for marked teammates to find a good run-away position based on Partial (approximate) Dominant Regions. This method divides the field into regions based on the players time of arrival (similar to a Voronoi diagram based on the distance of arrival), each of which shows an area that players can reach faster than others.

One year later, Gabel and Riedmiller [86] proposed the use of a Simultaneous Perturbation Stochastic Approximation (SPSA) combined with a RPROP learning technique (RSPSA) to overcome the opponent's offside trap by coordinated passing and player movements. The receiver of the pass that breaks the opponent's defense starts running in the correct direction at the right time, preferably being positioned right before the offside line while running at its maximal speed when the pass is executed.

4.3 Dynamic Positioning and Role Exchange

The Dynamic Positioning and Role Exchange (DPRE), and Dynamic Covering [1, 2, 4, 19, 20] was based on previous work from Stone et al. [49,50,79] which suggested the use of flexible agent roles with protocols for switching among them. The concept was extended and players may exchange their positionings and player types in the current formation if the utility of that exchange is positive for the team. Positioning exchange utilities are calculated using the distances from the player's present positions to their strategic positions and the importance of their positionings in the formation on that situation.

4.4 Setplay Framework

Robotic cooperation demands coordination at team level, requiring planning at different abstraction viewpoints and situations. Setplays are frequently used in many human team sports, e.g. rugby, basket- ball, handball, soccer and baseball. Certainly, there are considerable differences between robot soccer and standard sports, but Setplays were, even so, always expected to have a considerable impact on team-level coordination and cooperation.

Tactics and skills of robots are always improving, and thus opponent teams try to adapt to new playing patterns and react to them. It is thus convenient to be able to define Setplays, through freely editable configuration files, or even a generic Setplay graphical editor.

Such a cooperative plan can be described and shared in a standardized, generic and flexible language [16], which is then interpreted and executed at run-time. The main advantage is the writing of arbitrary Setplays, which are dynamically executed during the game, allowing the definition of new plays which could possibly differ between games, in order to adapt better to each opponent. Another benefit is the possibility to swiftly react to situations identified as advantageous: a new Setplay can quickly be edited and immediately executed. Setplays can also be used in different leagues [16].

FC Portugal team developed a complete framework for the representation, execution and evaluation of high level, flexible plans for agents playing robotic soccer. The framework was presented in [16, 17, 22] , defining a generic Setplay definition language and several associated tools.

To fulfill these requirements, one defined a standard language to define Setplays, which can later be interpreted by any player in any league. The basic concepts of soccer (moves, conditions, actions, skills) were given a clear yet comprehensive definition.

A Setplay is built upon Steps, with alternative transitions between them. Steps should be seen as intermediary states of the Setplay. Transitions between intermediary steps demand the execution of actions between the players. And there are also termination conditions. This language is more thoroughly described in [18, 22].

A library has been developed in C++ to ease the implementation of Setplays in any team. The Framework provides different tools: a parser for Setplay definition files and an engine to manage Setplay selection and execution. Thus, to apply Setplays to a new team, two tasks have to be done initially: implement the verification of conditions and the execution of the Framework actions. To actually use the Setplays, the team has to start the execution of the Setplay by instantiating its parameters, and regularly (i.e.: in every execution cycle) update the Setplay status, supplying ball and players positions.

The Setplay Framework was designed with the goal of being general, flexible, parameterizable and applicable to any robotic soccer league. A Setplay has a participant list, as well as an optional list of parameters. It also has a list of Steps, that represent intermediary states in the Setplay's execution. Transitions between steps entail the execution of specific actions, and can be restricted by Conditions, such as a player being in a specific field area, or having ball possession. Conditions can also be used to trigger Setplay successful ending or abortion.

A major issue in the usage of the Framework is how to achieve coordination between the robots when executing a Setplay. Naturally, a complex Setplay must follow several steps, and all participating players must be closely synchronized in order to achieve fruitful cooperation. A communication and synchronization policy was defined, in a straightforward and concise manner, since the 2D Simulation league has strict communication restrictions.

Each step will be led by the so-called lead player, which will normally be the player with ball possession, since it is the one taking the most important decisions, while manipulating the ball. This player is naturally not fixed throughout the Setplay, and will change from step to step. It will monitor the execution of the Setplay, instructing the other players on Setplay begin, step entry and transition choice. The entry into a new step, which is decided by the lead player in charge of the previous step, normally implies the change of the lead player.

FCPortugal team was the primary test-bed for the Setplays. Its code supplied the main building blocks for the application of Setplays: a stable state-of-the-world, considering own observations, information shared by other players, and prediction of actions' and interactions' effects; and a set of actions and skills that allows the easy mapping of actions as defined in the Setplay Framework to concrete executions in the 2D simulator. The Framework has also been applied to team CAMBADA , from the RoboCup Middle-size league, created by the Aveiro University [6,7]. This implementation is described in more detail in [18].

5 Conclusions and Future Work

Since the start of the RoboCup initiative, several coordination techniques were proposed that tackle core MAS coordination issues in simulated robotic soccer. The majority of these techniques has dealt with the problem of adequate player positioning, due to its impact on the successful execution of other actions (e.g. passing) during a match. In general, positioning techniques have evolved from reactive to more deliberative approaches, meaning that players now put the team's goals in front of his own because it is the only way for successful coordination to be achieved. Due to its complexity, this problem as been studied in more narrower scopes (e.g. defensive and offensive situations like opponent marking and ball passing respectively) with good results. However, situations where the number of teammates and opponents is uneven still don't seem to be adequately addressed by any of these.

Coordination technologies have evolved a lot since the start of RoboCup mostly due to added functionalities and constraints in the latest simulator releases. Although the use of communication and intelligent perception can assist team coordination through the sharing of pertinent world information and enhance the player's world state accuracy respectively, the simulator constraints discourage relying solely on them. Team strategies are usually very complex and are typically embedded into players knowledge prior to a game. The strategic approaches have also evolve from fixed policies to more flexible and dynamic policies that are based on real-time match information and previous opponent knowledge. Coaching was used to tweak team strategy mostly by giving advices to players and allow a quicker adaptation to opponent's behavior. Training methods have been used as a foundation to build into team members effective knowledge that can accelerate team coordination during real-time match situations (e.g. learning opponent behavior).

In order to succeed, a good coordination methodology should always consider the following aspects: Incorporate past knowledge to accelerate initial decisions for usual situations, driven from direct human expertise or by offline learned prediction models. This knowledge can be tailored for specific opponents; Knowledge should be adaptable according to opponent behavior in real-time; Use alternative techniques to complement and replace technologies based on communication and perception.

Setplays are a new concept that recently emerged on RoboCup. The Setplay Framework implemented by FC Portugal team has shown to be very effective, enabling to clearly outperform the original team's behavior, in similar circumstances. Since the Framework is presented as a stand-alone library, its usage is also quite simple: a new team wishing to use it only needs to define the domain specific concepts and deal with Setplay selection and instantiation. From this point on, it suffices to update the ball and players positions regularly to have Setplays executed.

Future work will be concerned with completely defining a common framework for cooperative robotics composed by: Strategy, Formations, DPRE, complete tactical/formation framework including graphical interface and complete setplay framework with a graphical interface for defining the setplays. This will be released as a library enabling to coordinate any RoboCup team in any league. Thus it will enable teams to research only in low level robotics in issues such as skills, vision, mechanics, kicking, among others, without any concern with the team coordination that will be managed by this framework.

Acknowledgements. This work was funded by the ERDF – European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through FCT - Portuguese Foundation for Science and Technology within project "INTELLWHEELS - Intelligent Wheelchair with Flexible Multimodal Interface" (FCT/RIPD/ADA/109636/2009) and project "ACORD: Adaptive Coordination of Robot Teams" (FCT-PTDC/EIA/70695/2006). The second author' was supported with a PROFAD scholarship from IPV.

References

1. Reis, L.P., Lau, N.: FC Portugal Team Description: RoboCup 2000 Simulation League Champion. In: Stone, P., Balch, T., Kraetzschmar, G.K. (eds.) RoboCup 2000. LNCS (LNAI), vol. 2019, pp. 29–40. Springer, Heidelberg (2001)
2. Reis, L.P., Lau, N., Oliveira, E.: Situation Based Strategic Positioning for Coordinating a Team of Homogeneous Agents. In: Hannebauer, M., Wendler, J., Pagello, E. (eds.) ECAI-WS 2000. LNCS (LNAI), vol. 2103, pp. 175–197. Springer, Heidelberg (2001)
3. Lau, N., Reis, L.P.: FC Portugal 2001 Team Description: Flexible Teamwork and Configurable Strategy. In: Birk, A., Coradeschi, S., Tadokoro, S. (eds.) RoboCup 2001. LNCS (LNAI), vol. 2377, pp. 515–518. Springer, Heidelberg (2002)
4. Lau, N., Reis, L.P.: FC Portugal - High-level Coordination Methodologies in Soccer Robotics. In: Lima, P. (ed.) Robotic Soccer, pp. 167–192. Itech Education and Publishing, Vienna (2007) ISBN 978-3-902613-21-9
5. Scolari Conceição, A., Moreira, A., Reis, L.P., Costa, P.J.: Architecture of Cooperation for Multi-Robot Systems. In: 1st IFAC/MVS 2006, Salvador, Brazil, October 2-3 (2006)
6. Lau, N., Lopes, L.S., Filipe, N., Corrente, G.: Roles, Positionings and Set Plays to Coordinate a RoboCup MSL Team. In: Lopes, L.S., Lau, N., Mariano, P., Rocha, L.M. (eds.) EPIA 2009. LNCS (LNAI), vol. 5816, pp. 323–337. Springer, Heidelberg (2009)
7. Lau, N., Lopes, L.S., Corrente, G., Filipe, N.: Multi-Robot Team Coordination Through Roles, Positioning and Coordinated Procedures. In: IROS 2009, St. Louis, USA (2009)
8. Shafii, N., Reis, L.P., Lau, N.: Biped Walking Using Coronal and Sagittal Movements Based on Truncated Fourier Series. In: Ruiz-del-Solar, J. (ed.) RoboCup 2010. LNCS, vol. 6556, pp. 324–335. Springer, Heidelberg (2010)
9. Picado, H., Gestal, M., Lau, N., Reis, L.P., Tomé, A.M.: Automatic Generation of Biped Walk Behavior Using Genetic Algorithms. In: Cabestany, J., Sandoval, F., Prieto, A., Corchado, J.M. (eds.) IWANN 2009, Part I. LNCS, vol. 5517, pp. 805–812. Springer, Heidelberg (2009)
10. Rei, L., Reis, L.P., Lau, N.: Optimizing a Humanoid Robot Skill. In: Lima, P., Cardeira, C. (eds.) Int. Conf. Mobile Robots and Competitions, Lisbon, Portugal, pp. 79–83 (2011)
11. Reis, L.P., Lau, N.: COACH UNILANG - A Standard Language for Coaching a (Robo)Soccer Team. In: Birk, A., Coradeschi, S., Tadokoro, S. (eds.) RoboCup 2001. LNCS (LNAI), vol. 2377, pp. 183–192. Springer, Heidelberg (2002)
12. Abreu, P.H., Moura, J., Silva, D.C., Reis, L.P., Garganta, J.: Performance Analysis in Soccer: a Cartesian Coordinates based Approach using RoboCup Data. Soft Computing 16, 47–61 (2012)

13. Faria, B.M., Reis, L.P., Lau, N., Castillo, G.: Machine Learning Algorithms applied to the Classification of Robotic Soccer Formations and Opponent Teams. In: IEEE CIS 2010, Singapore, pp. 344–349 (2010)
14. Almeida, R., Reis, L.P., Jorge, A.M.: Analysis and Forecast of Team Formation in the Simulated Robotic Soccer Domain. In: Lopes, L.S., Lau, N., Mariano, P., Rocha, L.M. (eds.) EPIA 2009. LNCS (LNAI), vol. 5816, pp. 239–250. Springer, Heidelberg (2009)
15. Lau, N., Reis, L.P., Certo, J.: Understanding Dynamic Agent's Reasoning. In: Neves, J., Santos, M.F., Machado, J.M. (eds.) EPIA 2007. LNCS (LNAI), vol. 4874, pp. 542–551. Springer, Heidelberg (2007)
16. Mota, L., Reis, L.P., Lau, N.: Multi-Robot Coordination using Setplays in the Middle-size and Simulation Leagues. Mechatronics 21(2), 434–444 (2011)
17. Mota, L., Reis, L.P., Lau, N.: Co-ordination in RoboCup's 2D Simulation League: Setplays as flexible, multi-robot plans. In: IEEE International Conference on Robotics, Automation and Mechatronics (RAM 2010), Singapore (2010)
18. Mota, L.: Multi-robot Coordination using Flexible Setplays: Applications in RoboCup's Simulation and Middle-Size Leagues. PhD thesis, Faculdade Eng. Univ. Porto (2012)
19. Reis, L.P., Lau, N.: FCPortugal Homepage, http://www.ieeta.pt/robocup
20. Reis, L.P.: Coordenação de Sistemas Multi-Agente: Aplicações na Gestão Universitária e no Futebol Robótico, PhD thesis, Faculdade de Engenharia da Univ. do Porto (2003)
21. Ferreira, R.A., Reis, L.P., Lau, N.: Situation Based Communication for Coordination of Agents. In: Reis, L.P., Moreira, A.P., Costa, E.S.P., Almeida, J.M. (eds.) Proc. Scientific Meeting of the Portuguese Robotics Open, Porto, FEUP Edições, pp. 39–44 (2004)
22. Mota, L., Reis, L.P.: Setplays: Achieving coordination by the appropriate use of arbitrary pre-defined flexible plans and inter-robot communication. In: Winfield, A.F.T., Redi, J. (eds.) ROBOCOMM 2007, vol. 318. ACM (2007)
23. Certo, J., Lau, N., Reis, L.P.: A Generic Strategic Layer for Collaborative Networks. In: Establ. Found. for Collaborative Networks. IFIP, vol. 243, pp. 273–282. Springer (2007)
24. Abdolmaleki, A., Movahedi, M., Salehi, S., Lau, N., Reis, L.P.: A Reinforcement Learning Based Method for Optimizing the Process of Decision Making in Fire Brigade Agents. In: Antunes, L., Pinto, H.S. (eds.) EPIA 2011. LNCS (LNAI), vol. 7026, pp. 340–351. Springer, Heidelberg (2011)
25. Almeida, F., Lau, N., Reis, L.P.: A Survey on Coordination Techniques for Simulated Robotic Soccer Teams. In: MAS&S@MALLOW 2010, Lyon, France (2010)
26. Portela, J., Abreu, P., Reis, L.P., Oliveira, E., Garganta, J.: An Intelligent Framework for Automatic Event Detection in Robotic Soccer Games: An Auxiliar Tool to Help Coaches Improving their Teams Performance. In: ICEIS 2010, pp. 244–249. INSTICC Press (2010)
27. Faria, B.M., Santos, B.S., Lau, N., Reis, L.P.: Data Visualization for Analyzing Simulated Robotic Soccer Games. In: IMAGAPP/IVAPP 2010, Angers, France, May 17-21, pp. 161–168. INSTICC Press (2010)
28. Faria, B.M., Castillo, G., Lau, N., Reis, L.P.: Classification of FC Portugal Robotic Soccer Formations: A Comparative Study of Machine Learning Algorithms Robotica Magazine. 1st Trimester, vol. 82, pp. 4–9 (2011) ISSN: 0874-9019
29. Almeida, P., Abreu, P.H., Lau, N., Reis, L.P.: Automatic Extraction of Goal-Scoring Behaviors from Soccer Matches. In: IROS 2012, Vilamoura, Portugal, pp. 849–856 (2012)
30. Abreu, P., Moreira, J., Costa, I., Castelão, D., Moreira, J., Reis, L.P., Garganta, J.: Human vs. Virtual Robotic Soccer: A Technical Analysis about Two Realities. European Journal of Sport Science 12(1), 26–35 (2012) ISSN: 1746-1391
31. Santiago, C., Sousa, A., Estriga, M.L., Reis, L.P., Lames, M.: Survey on Team Tracking Techniques applied to Sports. In: AIS 2010 (2010)
32. Abreu, P., Faria, M., Reis, L.P., Garganta, J.: Knowledge Representation in Soccer Domain: An Ontology Development. In: Rocha, A., Sexto, C.F., Reis, L.P., Cota, M.P. (eds.) CISTI 2010, Santiago de Compostela, Spain, June 16–19, pp. 414–419 (2010)

33. Santiago, C., Sousa, A., Reis, L.P., Estriga, M.L.: Real Time Colour Based Player Tracking in Indoor Sports. Computational Vision and Medical Image Processing. Computational Methods in Applied Sciences 19, 17–35 (2011)
34. Abreu, P., Costa, I., Castelão, D., Reis, L.P., Garganta, J.: Human vs. Robotic Soccer: How Far are They? A Statistical Comparison. In: Ruiz-del-Solar, J. (ed.) RoboCup 2010. LNCS (LNAI), vol. 6556, pp. 242–253. Springer, Heidelberg (2010)
35. Santiago, C., Sousa, A., Reis, L.P., Estriga, M.L.: Automatic Detection and Tracking of Handball Players. In: Tavares, J., Jorge, R.N. (eds.) VipImage 2009, October 14-16, pp. 213–219. Taylor & Francis Group, Porto (2009)
36. Domingues, E., Lau, N., Pimentel, B., Shafii, N., Reis, L.P., Neves, A.J.R.: Humanoid Behaviors: From Simulation to a Real Robot. In: Antunes, L., Pinto, H.S. (eds.) EPIA 2011. LNCS (LNAI), vol. 7026, pp. 352–364. Springer, Heidelberg (2011)
37. Mota, L., Reis, L.P.: A Common Framework for Co-operative Robotics: An Open, Fault Tolerant Architecture for Multi-league RoboCup Teams. In: Carpin, S., Noda, I., Pagello, E., Reggiani, M., von Stryk, O. (eds.) SIMPAR 2008. LNCS (LNAI), vol. 5325, pp. 171–182. Springer, Heidelberg (2008)
38. Mota, L., Lau, N., Reis, L.P.: Co-ordination in RoboCup's 2D Simulation League: Setplays as flexible, Multi-Robot plans. In: IEEE RAM 2010, Singapore, pp. 362–367 (2010)
39. Reis, L.P., Lopes, R., Mota, L., Lau, N.: Playmaker: Graphical Definition of Formations and Setplays. In: Rocha, A., Sexto, C.F., Reis, L.P., Cota, M.P. (eds.) CISTI 2010, Santiago de Compostela, Spain, June 16–19, pp. 582–587 (2010)
40. Mota, L., Reis, L.P.: An Elementary Communication Framework for Open Co-operative RoboCup Soccer Teams. In: Sapaty, P., Filipe, J. (eds.) ICINCO 2007/MARS 2007, Angers, France, May 9-12, pp. 97–101 (2007)
41. Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E., Matsubara, H.: RoboCup: A Challenge Problem for AI and Robotics. In: Kitano, H. (ed.) RoboCup 1997. LNCS, vol. 1395, pp. 1–19. Springer, Heidelberg (1998)
42. Chen, M., Foroughi, E., Heintz, F., Huang, Z., Kapetanakis, S., Kostiadis, K., Kummeneje, J., Noda, I., Obst, O., Riley, P., Steffens, T., Wang, Y., Yin, X.: RoboCup Soccer Server Manual (2007), `http://downloads.sourceforge.net/sserver/manual.pdf`
43. Sedaghat, M., et al.: Caspian 2003 Presentation Description. In: RoboCup 2003 (2003) (CD Proceedings)
44. Obst, O., Rollmann, M.: Spark – A Generic Simulator for Physical Multi-agent Simulations. In: Lindemann, G., Denzinger, J., Timm, I.J., Unland, R. (eds.) MATES 2004. LNCS (LNAI), vol. 3187, pp. 243–257. Springer, Heidelberg (2004)
45. Riley, P.: SPADES: System for Parallel Agent Discrete Event Simulation. AI Magazine 24(2), 41–42 (2003)
46. Smith, R.: Open Dynamics Engine v0.5 User Guide (2006), `http://opende.sourceforge.net/`
47. Obst, O., Rollmann, M.: Spark - A generic simulator for physical multi-agent simulations. Computer Systems: Science & Engineering 20(5) (2005)
48. McMillen, C., Veloso, M.: Distributed, Play-Based Coordination for Robot Teams in Dynamic Environments. In: Lakemeyer, G., Sklar, E., Sorrenti, D.G., Takahashi, T. (eds.) RoboCup 2006: Robot Soccer World Cup X. LNCS (LNAI), vol. 4434, pp. 483–490. Springer, Heidelberg (2007)
49. Stone, P.: Layered Learning in Multiagent Systems: A Winning Approach to Robotic Soccer. MIT Press (2000) ISBN: 0262194384
50. Stone, P., Veloso, M.: Task Decomposition, Dynamic Role Assignment, and Low-Bandwidth Communication for Real-Time Strategic Teamwork. Artificial Intelligence 110(2), 241–273 (1999)

51. Kok, J.R., Spaan, M.T.J., Vlassis, N.: An approach to noncommunicative multiagent coordination in continuous domains. In: Wiering, M. (ed.) Proc. of the 12th Belgian-Dutch Conference on Machine Learning, Utrecht, The Netherlands, December 2002, pp. 46–52 (2002)
52. Kok, J.R., Spaan, M.T.J., Vlassis, N.: Multi-Robot Decision Making using Coordination Graphs. In: Almeida, A.T., Nunes, U. (eds.) Proc. of the 11th International Conference on Advanced Robotics, Coimbra, Portugal, pp. 1124–1129 (2003)
53. Kok, J.R., Spaan, M.T.J., Vlassis, N.: Non-Communicative Multi-Robot Coordination in Dynamic Environments. Robotics and Autonomous Systems 50(2-3), 99–114 (2005)
54. Vail, D., Veloso, M.: Multi-Robot Dynamic Role Assignment and Coordination Through Shared Potential Fields. In: Schultz, A., et al. (eds.) Multi-Robot Systems. Kluwer (2003)
55. Isik, M., Stulp, F., Mayer, G., Utz, H.: Coordination Without Negotiation in Teams of Heterogeneous Robots. In: Lakemeyer, G., Sklar, E., Sorrenti, D.G., Takahashi, T. (eds.) RoboCup 2006: Robot Soccer World Cup X. LNCS (LNAI), vol. 4434, pp. 355–362. Springer, Heidelberg (2007)
56. Stulp, F., Isik, M., Beetz, M.: Implicit Coordination in Robotic Teams using Learned Prediction Models. In: ICRA 2006, pp. 1330–1335. IEEE, Orlando (2006)
57. Stulp, F., Utz, H., Isik, M., Mayer, G.: Implicit Coordination with Shared Belief: A Heterogeneous Robot Soccer Team Case Study. Advanced Robotics 24(7), 1017–1036 (2010)
58. Desheng, X., Kejian, X.: Role Assignment, Non-Communicative Multi-agent Coordination in Dynamic Environments Based on the Situation Calculus. In: Proc. of the WRI Global Congress on Intelligent Systems, vol. 1, pp. 89–93 (2009)
59. Cheny, M., Dorer, K., Foroughi, E., Heintz, F., Huangy, Z., Spiros, S., Kostiadis, K., Kummeneje, J., Murray, J., Noda, I., Obst, O., Riley, P., Stevens, T., Wangy, Y., Yiny, X.: RoboCup Soccer Server Users Manual. Version 7.07 and later. RoboCup Federation (2003)
60. Buttinger, S., Diedrich, M., Hennig, L., Hoenemann, A., Huegelmeyer, P., Nie, A., Pegam, A., Rogowski, C., Rollinger, C., Steffens, T., Teiken, W.: The Dirty Dozen Team and Coach Description. In: Birk, A., Coradeschi, S., Tadokoro, S. (eds.) RoboCup 2001. LNCS (LNAI), vol. 2377, pp. 543–546. Springer, Heidelberg (2002)
61. Davin, J., Riley, P., Veloso, M.: CommLang: Communication for Coachable Agents. In: Nardi, D., Riedmiller, M., Sammut, C., Santos-Victor, J. (eds.) RoboCup 2004. LNCS (LNAI), vol. 3276, pp. 46–59. Springer, Heidelberg (2005)
62. Scerri, P., Pynadath, D.: Tambe. M.: Adjustable autonomy in real-world multi-agent environments. In: Proc of the 5th International Conference on Autonomous Agents, AGENTS 2001, pp. 300–307. ACM, New York (2005) ISBN 1-58113-326-X
63. Riley, P., Veloso, M., Kaminka. G.: An Empirical Study of Coaching. In: Asama, H., Arai, T., Fukuda, T., Hasegawa, T. (eds.) Distributed Autonomous Robotic Systems 5, pp. 215–224. Springer (2002)
64. Takahashi, T.: Kasugabito III. In: Veloso, M.M., Pagello, E., Kitano, H. (eds.) RoboCup 1999. LNCS (LNAI), vol. 1856, pp. 592–595. Springer, Heidelberg (2000)
65. Riley, P.: Veloso. M.: On Behavior Classification in Adversarial Environments. In: Procs. of the 5th International Symposium on Distributed Autonomous Robotic Systems (2000)
66. Visser, U., Drücker, C., Hübner, S., Schmidt, E., Weland, H.-G.: Recognizing Formations in Opponent Teams. In: Stone, P., Balch, T., Kraetzschmar, G.K. (eds.) RoboCup 2000. LNCS (LNAI), vol. 2019, pp. 391–396. Springer, Heidelberg (2001)
67. Riley, P., Veloso, M.: Recognizing Probabilistic Opponent Movement Models. In: Birk, A., Coradeschi, S., Tadokoro, S. (eds.) RoboCup 2001. LNCS (LNAI), vol. 2377, pp. 453–540. Springer, Heidelberg (2002)
68. Riley, P., Veloso, M., Kaminka, G.: An Empirical Study of Coaching. In: Asama, H., Arai, T., Fukuda, T., Hasegawa, T. (eds.) Distributed Autonomous Robotic Systems 5, pp. 215–224. Springer (2002)
69. Ahmadi, M., Lamjiri, A., Nevisi, M., Habibi, J., Badie, K.: Using a two-layered case-based reasoning for prediction in soccer coach. In: Arabnia, H., Kozerenko, E. (eds.) ICML: Models, Techn. and Applications, pp. 181–185. CSREA Press, Las Vegas (2003)

70. Kuhlmann, G., Stone, P., Lallinger, J.: The UT Austin Villa 2003 Champion Simulator Coach: A Machine Learning Approach. In: Nardi, D., Riedmiller, M., Sammut, C., Santos-Victor, J. (eds.) RoboCup 2004. LNCS (LNAI), vol. 3276, pp. 636–644. Springer, Heidelberg (2005)
71. Stolzenburg, F., Murray, J., Sturm, K.: Multiagent Matching Algorithms With and Without Coach. Decision Systems 15(2-3), 215–240 (2006)
72. Castelpietra, C., Guidotti, A., Iocchi, L., Nardi, D., Rosati, R.: Design and Implementation of Cognitive Soccer Robots. In: Birk, A., Coradeschi, S., Tadokoro, S. (eds.) RoboCup 2001. LNCS (LNAI), vol. 2377, pp. 312–318. Springer, Heidelberg (2002)
73. Iocchi, L., Marchetti, L., Nardi, D., Lima, P., Barbosa, M., Pereira, H., Lopes, N.: SPQR + ISocRob RoboCup 2007 qualification report. Technical report, Sapienza Universitá di Roma, Italy; Instituto Superior Técnico (2007)
74. Kok, J., Vlassis, N.: Collaborative Multiagent Reinforcement Learning by Payoff Propagation. Journal of Machine Learning Research (JMLR) 7, 1789–1828 (2006)
75. Ruiz, M., Uresti, J.: Team Agent Behavior Architecture in Robot Soccer. In: Procs of the Latin American Robotic Symposium, pp. 20–25 (2008)
76. Vega, J.L., Junco, M.A., Ramírez, J.: Major behavior definition of football agents through XML. In: Arai, T., Pfeifer, R., Balch, T., Yokoi, H. (eds.) Procs. of the 9th International Conference on Intelligent Autonomous Systems, University of Tokyo, Tokyo, Japan, pp. 668–675. IOS Press (2006)
77. Dashti, H.T., Aghaeepour, N., Asadi, S., Bastani, M., Delafkar, Z., Disfani, F.M., Ghaderi, S.M., Kamali, S., Pashami, S., Siahpirani, A.F.: Dynamic Positioning Based on Voronoi Cells (DPVC). In: Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y. (eds.) RoboCup 2005. LNCS (LNAI), vol. 4020, pp. 219–229. Springer, Heidelberg (2006)
78. Akiyama, H., Noda, I.: Multi-agent Positioning Mechanism in the Dynamic Environment. In: Visser, U., Ribeiro, F., Ohashi, T., Dellaert, F. (eds.) RoboCup 2007: Robot Soccer World Cup XI. LNCS (LNAI), vol. 5001, pp. 377–384. Springer, Heidelberg (2008)
79. Stone, P., Veloso, M., Riley, P.: The CMUnited-98 Champion Simulator Team. In: Asada, M., Kitano, H. (eds.) RoboCup 1998. LNCS (LNAI), vol. 1604, pp. 61–76. Springer, Heidelberg (1999)
80. Kyrylov, V., Hou, E.: While the Ball in the Digital Soccer is Rolling, Where the Non-Player Characters Should go in a Defensive Situation? In: Kapralos, B., Katchabaw, M., Rajnovich, J. (eds.) ACM Proc. Conf. on Future Play, Toronto, Canada, pp. 90–96 (2007)
81. Kyrylov, V., Catalina, M., Ng, H.: Rollover as a Gait in Legged Autonomous Robots: A Systems Analysis. In: Baltes, J., Lagoudakis, M.G., Naruse, T., Ghidary, S.S. (eds.) RoboCup 2009. LNCS (LNAI), vol. 5949, pp. 166–178. Springer, Heidelberg (2010)
82. Gabel, T., Riedmiller, M., Trost, F.: A Case Study on Improving Defense Behavior in Soccer Simulation 2D: The NeuroHassle Approach. In: Iocchi, L., Matsubara, H., Weitzenfeld, A., Zhou, C. (eds.) RoboCup 2008. LNCS (LNAI), vol. 5399, pp. 61–72. Springer, Heidelberg (2009)
83. Kyrylov, V., Razykov, S.: Pareto-Optimal Offensive Player Positioning in Simulated Soccer. In: Visser, U., Ribeiro, F., Ohashi, T., Dellaert, F. (eds.) RoboCup 2007: Robot Soccer World Cup XI. LNCS (LNAI), vol. 5001, pp. 228–237. Springer, Heidelberg (2008)
84. Razykov, S., Kyrylov, V.: While the Ball in the Digital Soccer Is Rolling, Where the Non-Player Characters Should Go If the Team is Attacking? In: Proc. ACM Conference on Future Play, London, Ontario, Canada (2006)
85. Nakanishi, R., Murakami, K., Naruse, T.: Dynamic Positioning Method Based on Dominant Region Diagram to Realize Successful Cooperative Play. In: Visser, U., Ribeiro, F., Ohashi, T., Dellaert, F. (eds.) RoboCup 2007: Robot Soccer World Cup XI. LNCS (LNAI), vol. 5001, pp. 488–495. Springer, Heidelberg (2008)
86. Gabel, T., Riedmiller, M.: Brainstormers 2D - Team Description (2009)

Part I
Artificial Intelligence

Hierarchical Plan-Based Control in Open-Ended Environments: Considering Knowledge Acquisition Opportunities

Dominik Off and Jianwei Zhang

TAMS, University of Hamburg, Vogt-Koelln-Strasse 30, Hamburg, Germany
{off,zhang}@informatik.uni-hamburg.de
http://tams-www.informatik.uni-hamburg.de/

Abstract. We introduce a novel hierarchical planning approach that extends previous approaches by additionally considering decompositions that are only applicable with respect to a consistent extension of the (open-ended) domain model at hand. The introduced planning approach is integrated into a plan-based control architecture that interleaves planning and execution automatically so that missing information can be acquired by means of active knowledge acquisition. If it is more reasonable, or even necessary, to acquire additional information prior to making the next planning decision, the planner postpones the overall planning process, and the execution of appropriate knowledge acquisition tasks is automatically integrated into the overall planning and execution process.

Keywords: Plan-based Control, Continual Planning, HTN Planning, Reasoning, Knowledge Representation, Plan Execution.

1 Introduction

Planning their future course of action is particularly difficult for agents (e.g., robots) that act in a *dynamic* and *open-ended* environment where it is unreasonable to assume that a complete representation of the state of the domain is available. We define an *open-ended domain* as a domain in which agents can in general neither be sure of having all information nor of knowing all possible states (e.g., all objects) of the world they inhabit.

Conformant, *contingent* or *probabilistic* planning approaches can be used to generate plans in situations where insufficient information is available at planning time [13,4]. These approaches generate conditional plans—or policies—for all possible contingencies. Unfortunately, these approaches are computationally hard, scale badly in dynamic unstructured domains and are only applicable if it is possible to foresee all possible outcomes of a knowledge acquisition process [12,8]. Therefore, these approaches can hardly be applied to the dynamic and open-ended domains we are interested in. Consider, for example, a robot agent that is instructed to bring Bob's mug into the kitchen, but does not know the location of the mug. Generating a plan for all possible locations in a three dimensional space obviously is unreasonable and practically impossible.

A more promising approach for agents that act in open-ended domains is *continual planning* [2] which enables the interleaving of planning and execution so that missing

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 25–38, 2013.

information can be acquired by means of active information gathering. Existing continual planning systems can deal with incomplete information. However, they usually rely on the assumption that all possible states of a domain are known. This makes it, for example, difficult to deal with a priori unknown object instances. Another important issue that is not directly considered by previous work is the fact that a knowledge acquisition task $task_1$ can—like any other task—make the execution of an additional knowledge acquisition task $task_2$ necessary which might require the execution of the knowledge acquisition task $task_3$ and so on. Consider, for example, a situation where a robot is instructed to deliver Bob's mug into Bob's office. Moreover, let us assume that the robot does know that Bob's mug is in the kitchen, but does not know the exact location of the mug. In this situation, the robot needs to perform a knowledge acquisition task that determines the exact location of Bob's mug. However, in order to do that via perception the robot first needs to go into the kitchen. If the robot does not have all necessary information in order to plan how to get into the kitchen (e.g., it is unknown whether the kitchen door is open or closed), then it needs to first perform additional knowledge acquisition tasks that acquire this information. Existing continual planning approaches usually fail to cope with such a situation. In contrast, we propose a continual planning and acting approach that is able to deal with these kind of situations and thus can enable an agent to perform tasks in a larger set of situations.

We are trying to give an answer to the following questions: How can an agent determine knowledge acquisition activities that make it possible to find a plan when necessary information is missing? When is it more reasonable to acquire additional information prior to continuing the planning process? How to automatically switch between planning and acting?

The main contributions of this work are:

- to propose the new HTN planning system *ACogPlan* that additionally considers planning alternatives that are possible with respect to a consistent extension of the domain model at hand, and is able to autonomously decide when it is more reasonable to acquire additional information prior to continuing the planning process;
- to propose the ACogPlan-based, high-level control system *ACogControl* that enables an agent to perform tasks in open-ended domains;
- and to present a set of experiments that demonstrate the performance characteristics of the overall approach.

2 HTN Planning in Open-Ended Domains

In this section, we present the continual HTN planning system ACogPlan. We describe the planning phase of the overall plan-based control system.

2.1 General Idea

The proposed planning system ACogPlan can be seen as an extension of the SHOP [10] forward search (i.e., forward decomposition) *Hierarchical Task Network (HTN)* planning system. The SHOP algorithm generates plans by successively choosing an instance of a *relevant*[1] HTN method or planning operator for which an instance of the

[1] As defined in [4, Definition 11.4].

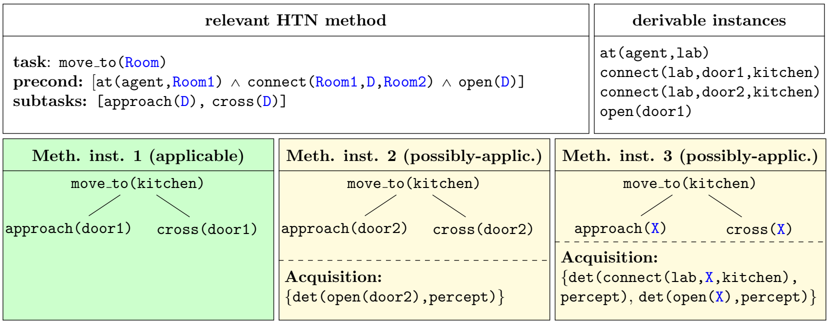

relevant HTN method
task: move_to(Room)
precond: [at(agent,Room1) ∧ connect(Room1,D,Room2) ∧ open(D)]
subtasks: [approach(D), cross(D)]
derivable instances
at(agent,lab)
connect(lab,door1,kitchen)
connect(lab,door2,kitchen)
open(door1)
Meth. inst. 1 (applicable)
move_to(kitchen)
approach(door1)
cross(door1)
Meth. inst. 2 (possibly-applic.)
move_to(kitchen)
approach(door2)
cross(door2)
Acquisition:
{det(open(door2),percept)}
Meth. inst. 3 (possibly-applic.)
move_to(kitchen)
approach(X)
cross(X)
Acquisition:
{det(connect(lab,X,kitchen),
percept), det(open(X),percept)}

Fig. 1. Possible method instances for the task move_to(kitchen)

precondition can be derived with respect to the domain model at hand. However, in open-ended domains it will often be possible to instantiate additional HTN methods or planning operators (i.e., which precondition is not derivable) if additional information is available. The general idea of the proposed planning system *ACogPlan* is to also consider instances of relevant HTN methods and planning operators for which the precondition cannot be derived but might be derivable with respect to a consistent extension of the domain model (i.e., if additional information is available).

For example, consider a simple situation where a robot is instructed to perform the task move_to(kitchen) as illustrated by Fig. 1.[2] In this situation there is only one relevant HTN method. It is known that the robot is in the lab, the lab is connected to the kitchen via door1 and door2, and door1 is open. For the illustrated example, existing HTN planners would only consider the first instance of the relevant HTN method that plans to approach and cross door1. The proposed HTN planning algorithm *ACogPlan*, however, also considers two additional instances of the relevant HTN method which cannot directly be applied, but are applicable in a consistent extension of the given domain. Methods or planning operators that are only applicable with respect to an extension of an agent's domain model are called *possibly-applicable*. For example, it will also be possible to cross door2 if the robot could find out that this door is open. Moreover, in open-ended domains it can also be possible that there is another door which connects the lab and the kitchen.

Additionally considering possibly-applicable HTN methods or planning operators is important in situations where one cannot assume that all information is available at the beginning of the planning process. It often enables the generation—and execution—of additional plans. In particular, it can enable a planner to generate plans where it would otherwise be impossible to generate any plan at all. For example, if it were unknown whether door1 is open or closed, then there would only be possibly-applicable method instances. Hence, without considering possible-applicable method instances a planner would fail to generate a plan for the task move_to(kitchen) and thus the agent would be unable to achieve its goals. Moreover, if the optimal plan requires knowledge acquisition, then the optimal plan can only be found if possibly-applicable method and

[2] Please note that in the context of this work variables will be written as alphanumeric identifiers beginning with capital letters.

planning operator instances are considered. In other words, one can also benefit from the proposed approach in situations where it is possible to generate a complete plan without acquiring additional information.

2.2 Open-Ended Domain Model

A planner that wants to consider possibly-applicable HTN methods or planning operators needs to be able to reason about extensions of its domain model. Most existing automated planning systems are unable to do that, since their underlying domain model is based on the assumption that all information is available at the beginning of the planning process [9]. In contrast, the proposed HTN planning system *ACogPlan* is based on the open-ended domain model *ACogDM*. ACogDM enables the planner to reason about relevant extensions of its domain model. The key concepts of ACogDM are described briefly in this section.

A planner should only consider domain model extensions that are *possible* and *relevant* with respect to the overall task. However, how can a planner infer what is relevant and possible? The domain information encoded in HTN methods can nicely be exploited in order to infer which information is relevant. A relevant method or planning operator can actually be applied if and only if its precondition p holds (i.e., an instance $p\sigma$[3] is derivable) with respect to the given domain model. Therefore, we define the set of *relevant preconditions* with respect to a given *planning context* (i.e., a domain model and a task list) to be the set of all preconditions of relevant methods or planning operators. An HTN planner cannot—except backtracking—continue the planning process in situations where no relevant precondition is derivable with respect to the domain model at hand. Introducing the notion of a relevant precondition is a first step to determine relevant extensions of a domain model, since only domain model extensions that make the derivation of an additional instance of a relevant precondition possible constitute an additional way to continue the planning process. All other possible extensions are irrelevant, because they do not imply additional planning alternatives. In other words, if it is possible to acquire additional information which implies the existence of a new instance of a relevant precondition, then the planning process can be continued in an alternative manner. As already pointed out, this is particularly relevant for situations in which it would otherwise be impossible to find any plan at all.

In order to formalize this we introduce the following concepts: a *possibly-derivable statement* (e.g., a precondition) and an *open-ended literal*. Let L_x be a set of literals and p be a precondition. p is called *possibly-derivable* w.r.t. L_x iff the existence of a new instance $l\sigma$ for all $l \in L_x$ implies the existence of a new instance $p\sigma$ of p. Obviously this definition is only useful if the existence of an additional instance for each $l \in L_x$ is possible. A literal for which the existence of non-derivable instances is possible is called *open-ended*. Based on that, one can say that a possibly-derivable precondition constitutes the partition of a precondition into a derivable and an open-ended part (i.e., a set of open-ended literals).

For example, consider the situation illustrated by Fig. 1. In this example there are three different situations in which the precondition of the HTN method is possibly-derivable.

[3] In the context of this work σ denotes a substitution.

In all cases Room1 is substituted with lab and Room2 is substituted with kitchen. Furthermore, in the first situation D is substituted with door1 and the precondition is possibly-derivable with respect to the agents domain model and the set of open-ended literals $\{\}$. In the second case, D is substituted with door2 and the precondition is possibly-derivable with respect to the set of open-ended literals {open(door2)}. In the last case, D is not instantiated and the precondition is possibly-derivable with respect to the set of open-ended literals {connect(lab,D,kitchen), open(D)}. Thus, in this example the open-ended domain model ACogDM can tell the robot agent that it can cross door1, or cross door2 if it can find out that door2 is open, or cross another door D if it finds another door D that connects the lab and the kitchen and is open. In this way, ACogDM can enable a planner to reason about possible and relevant extensions of its domain model.

2.3 Planning Algorithm

In this section, we present the key conceptualizations and the algorithm of the proposed planning system.

Preliminaries. Dependencies between open-ended literals need to be considered by the generation of knowledge acquisition plans. For example, for the set of open-ended literals {mug(X),color(X,red)} one cannot independently acquire an instance of mug(X) and an instance of color(X,red), because one needs to find an instance of X which represents a mug as well as a red object. Let l_1, l_2 be literals that are part of a precondition p in disjunctive normal form and $var(l)$ denote the set of variables of a literal l. l_1 and l_2 are called *dependent* (denoted as $l_1 \leftrightarrow l_2$) iff l_1 and l_2 are part of the same conjunctive clause and $((var(l_1) \cap var(l_2) \neq \emptyset)$, or l_1 and l_2 are identical, or $(\exists_{l_3}\, l_1 \leftrightarrow l_3 \wedge l_3 \leftrightarrow l_2))$.

Agents (e.g., robots) can usually acquire information from a multitude of sources. These sources are called *external knowledge sources*. While submitting questions to external databases or reasoning components might be "simply" achieved by calling external procedures, submitting questions to other sources (e.g., perception), however, involves additional planning and execution. For the purpose of enabling ACogPlan to generate knowledge acquisition plans, we use a particular kind of task, namely a *knowledge acquisition task*. A Knowledge acquisition task has the form det(l, I, C, ks) where l is a literal, I is the set of all derivable instances of l, C is a set of literals that are dependent on l, and ks is a knowledge source. In other words, det(l, I, C, ks) is the task of acquiring an instance $l\sigma$ of l from the knowledge source ks such that $l\sigma \notin I$ (i.e., $l\sigma$ is not already derivable) and for all $c \in C$ an instance of $c\sigma$ is derivable. For example, det(open(kitchen_door),$\emptyset$,$\emptyset$,percept) is the task of determining whether the kitchen door is open by means of perception. Furthermore, det(mug(X), [mug(bobs_mug)],[in_room(X,r1),red(X)],hri(bob)) constitutes the task of finding a red mug which is located in the room r1 and is not Bob's mug by means of human robot interaction with Bob. Like for other tasks, we can define HTN methods that describe how to perform a knowledge acquisition task. For example, Fig. 2 shows a simple method for the acquisition task of determining whether a door is

```
method(det(open(D),I,C,percept),
(door(D)),                               % precondition
[approach(D), sense(open(D), percept)], % subtasks
50).                                     % cost
```

Fig. 2. Example HTN method for an acquisition task

open. Every method has an expected cost that describes how expensive it is to perform a task as described by the method. In this example, the cost is “hard-coded”, but it is also possible to calculate a situation dependent cost.

Knowledge acquisition tasks enable the planner to reason about possible knowledge acquisitions, since they describe (1) what knowledge acquisitions are possible under what conditions, (2) how expensive it is to acquire information from a specific knowledge source, and (3) how to perform a knowledge acquisition task.

It might be possible that the same information can be acquired from different external knowledge sources, and the expected cost to acquire the same information can be completely different for each source. Thus, in order to acquire additional instances for each literal of a set of open-ended literals, a planner needs to decide for each literal from which knowledge source it should try to acquire an additional instance. The result of this decision process is called a *knowledge acquisition scheme*. A knowledge acquisition scheme is a set of tuples (l, ks) where l is a literal and ks is an external knowledge source. It represents one possible combination of trying to acquire a non-derivable instance for each open-ended literal by an adequate knowledge source. For example, $\{$`(on_table(bobs_mug),percept), (white_coffee(bob),hri(bob))`$\}$ represents the fact that the query `on_table(bobs_mug)?` should be answered by perception and the query `white_coffee(bob)?` should be submitted to Bob. Formally a knowledge acquisition scheme is defined as follows:

Definition 1 (Knowledge Acquisition Scheme). *Let st be a statement that is possibly-derivable with respect to D_M and the set of open-ended literals $L_x = \bigcup_{1 \leq i \leq n} \{l_i\}$. Moreover let KS be the set of knowledge sources. A set $kas := \bigcup_{1 \leq i \leq n} \{(l_i, k_i)\}$ $(k_i \in \mathcal{KS})$ is called a* knowledge acquisition scheme *for st w.r.t. D_M. If $L_X = \emptyset$, then the corresponding knowledge acquisition scheme is also $\emptyset$.*

However, a knowledge acquisition scheme is only helpful for an agent if it is actually able to perform the corresponding knowledge acquisition tasks. For example, if a robot in principle is not able to find out whether a door is open, then the planner does not have to consider method instance 2 and 3 for the situation illustrated by Fig. 1. A knowledge acquisition scheme for which all necessary knowledge acquisition tasks can be possibly performed by the agent is called *possibly-acquirable* and more formally defined as follows:

Definition 2 (Possibly-acquirable). *An acquisition (l, ks) is called* possibly-acquirable *w.r.t. to a domain Model D_M iff there is an applicable or possibly-applicable planning step (i.e., a method or an operator) for the knowledge acquisition task $det(l, I, C, ks)$ such that I are all derivable instances of l w.r.t. D_M and C is the context. Moreover, a* *knowledge acquisition scheme kas is called possibly-acquirable iff all $(l, ks) \in kas$ are possibly-acquirable.*

Let $\mathcal{D}$ be the set of domain models, $\mathcal{TL}$ be the set of task lists, $\mathcal{P}$ be the set of plans and $\mathcal{KAS}$ be the set of knowledge acquisition schemes. We call $ps \in \mathcal{D} \times \mathcal{TL} \times \mathcal{P} \times \mathcal{KAS}$ a *planning state*. A planning state is called *final* if the task list is empty and called *intermediate* if the task list is not empty. ps_D denotes the domain model, ps_t the task list, ps_p the plan and ps_{kas} the knowledge acquisition scheme of a planning state ps.

The term *planning step* is used in this work as an abstraction of (HTN) methods and planning operators. A planning step $\mathfrak{s}$ is represented by a 4-tuple $(\mathfrak{s}_{task}, \mathfrak{s}_{cond}, \mathfrak{s}_{eff}, \mathfrak{s}_{cost})$. $\mathfrak{s}_{task}$ is an atomic formula that describes for which task $\mathfrak{s}$ is relevant, $\mathfrak{s}_{cond}$ is a statement that constitutes the precondition of $\mathfrak{s}$, $\mathfrak{s}_{eff}$ is the effect of $\mathfrak{s}$, and $\mathfrak{s}_{cost}$ represents the expected cost of the plan that results from the application of $\mathfrak{s}$.

Let PS be the set of planning states. $\mathfrak{s}_{eff}$ is a function $\mathfrak{s}_{eff} : PS \rightarrow PS$. Thus, a planning step maps the current planning state to a resulting planning state. In this sense operators map the current planning state to a resulting state by removing the next task from the task list, adding a ground instance of this task to the plan and updating the domain model according to the effects of the operator. In contrast, HTN methods transform the current planning state by replacing an active task by a number of subtasks.

Furthermore, we define the concept of a *possibly-applicable* planning step introduced in Section 2.1 as follows:

Definition 3 (Possibly-applicable). *A planning step $\mathfrak{s}$ is called* possibly-applicable *w.r.t. a domain model D_M and a knowledge acquisition scheme kas iff kas is possibly-acquirable and a knowledge acquisition scheme for $\mathfrak{s}_{cond}$.*

A possibly-applicable planning step can only be applied after necessary information has been acquired by the execution of corresponding knowledge acquisition tasks. For example, consider the second method instance of the situation illustrated by Fig. 1. This method instance can only be applied if the robot has perceived that `door2` is open. The fact that possibly-applicable planning step instances require the execution of additional tasks (i.e., knowledge acquisition tasks) needs to be considered by the expected cost. The cost of a possibly-applicable planning step is defined as the sum of the cost for the step if it is applicable and the expected cost of all necessary knowledge acquisition tasks.

For example, let us assume that the cost of the plan that results from applying the method for `move_to(Room)` is always 100. Moreover, let us assume that the cost of performing the task `det(open(door2),∅,∅,percept)` is 50 (see Fig. 2) and the cost of performing the task `det(connect(lab,X,kitchen), [connect(lab,door1,kitchen), connect(lab, door2,kitchen)], open(X), percept)` is 300. In this situation the cost of method instance 1 is 100, the cost of method instance 2 is $100 + 50 = 150$, and the cost of method instance 3 is $100 + 50 + 300 = 450$. Thus, in this case the applicable instance has the less expected cost. However, this does not always have to be the case.

```
   Result: a planning state $ps'$, or failure
1  if $ps$ is a final planning state then
2      return $ps$;
3  $steps \leftarrow \{(\mathfrak{s}, \sigma, kas) | \mathfrak{s}$ is the instance of a planning step, $\sigma$ is a substitution such
   that $\mathfrak{s}\sigma$ is relevant for the next task, $\mathfrak{s}$ is applicable or possibly-applicable w.r.t.
   $ps_D$ and the knowledge acquisition scheme $kas\}$;
4  if choose $(\mathfrak{s}, \sigma, kas) \in steps$ with the minimum overall cost then
5      if $kas = \emptyset$ then
6          $ps' \leftarrow \mathfrak{s}_{eff}(ps)$;
7          $ps'' \leftarrow$ plan($ps'$);
8          if $ps'' \neq failure$ then
9              return $ps''$;
10     else
11         return $(ps_D, ps_t, ps_p, kas)$;
12 else
13     return failure;
```

Fig. 3. Algorithm `plan`(ps)

Algorithm. The simplified algorithm of the proposed HTN planning system is shown by Fig. 3. The algorithm is an extension of the SHOP [10] algorithm that additionally considers possibly-applicable decompositions.

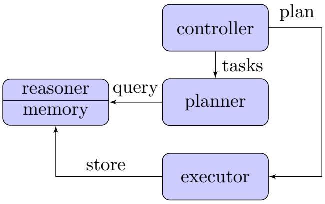
A planning state is the input of the recursive planning algorithm. If the task list of the given planning state is empty, then the planning process successfully generated a complete plan and the given planning state is returned. Otherwise, the algorithm successively chooses the applicable or possibly-applicable step with the lowest expected cost. If the planner chooses an applicable planning step (i.e., no knowledge acquisition is necessary and the knowledge acquisition scheme is the empty set), then it applies the step and recursively calls the planning algorithm with the updated planning state (line 5-9).

In contrast, if the planner chooses an only possibly-applicable planning step, then it stops the planning process and returns the current (intermediate) planning state including the knowledge acquisition scheme of the chosen planning step (line 10-11). In this way, the planner automatically decides whether it is more reasonable to continue the planning or to first acquire additional information. In other words, it decides when to switch between planning and acting. If it is neither possible to continue the planning process nor to acquire relevant information, then the planner backtracks to the previous choice point or returns *failure* if no such choice point exists.

3 Continual Planning and Acting

The overall idea of the proposed continual planning and acting system is to interleave planning and acting so that missing information can be acquired by means of active information gathering. In Section 2 we described a new HTN planning system for open-ended domains. Based on that, we describe the high-level control system ACogControl in this section.

The overall architecture is sketched in Fig. 4(a). The central component in this architecture is the *controller*. When the agent is instructed to perform a list of tasks, then this list is sent to the controller. The controller calls the planner described in Section 2 and decides what to do in situations where the planner only returns an intermediate planning state. Furthermore, the controller invokes the *executor* in order to execute—complete or partial—plans. The executor is responsible for the execution and execution monitoring of actions. In order to avoid unwanted loops (e.g., perform similar tasks more than once) it is essential to store relevant information of the execution process in the memory system. The executor stores information about the executed actions and the outcome of a sensing action in the memory system such that the domain model can properly be updated. This information includes acquired information as well as knowledge acquisition attempts. Knowledge acquisition attempts are stored to avoid submitting the same query more than once to a certain knowledge source.



(a) Illustration of the plan-based control architecture.

```
1  $ps \leftarrow$create-intial-ps($tasks$);
2  $ps' \leftarrow$ plan($ps$);
3  if $ps'$ is a final planning state then
4      $r \leftarrow$ execute($ps'_p$);
5      return $r$;
6  else
7      $r \leftarrow$perform($p' \subseteq ps'_p$);
8      if $r$ is a success then
9          choose $ac \in ps'_{kas}$ with the minimum cost;
10         $t_{ac} \leftarrow$ acquisition-task($ac$);
11         perform($[t_{ac}]$);
12     perform($overall\_tasks$);
```

(b) Algorithm perform($tasks$,$overall\_tasks$).

Fig. 4.

The behavior of the controller is specified by the algorithm shown at Fig. 4(b). When the controller is invoked, it first constructs an initial planning state based on the given task list and invokes the planner (lines 1-2). If the planner returns a final planning state (i.e., a planning state that contains a complete plan), then the controller directly forwards the generated plan to the executor.

However, if the planner returns an intermediate planning state (i.e., a planning state that only contains a partial plan), then the controller performs a prefix of the already generated plan, chooses the knowledge acquisition with the minimum expected cost, performs the corresponding knowledge acquisition task, and continues to perform the overall tasks. Please note that knowledge acquisition tasks can also require it to perform additional knowledge acquisition tasks. Furthermore, it usually cannot generically be determined which part of the already generated plan should be executed. For example, if we instruct a robot agent to deliver a cup into the kitchen, but it is unknown whether the door of the kitchen is open or closed, then it is reasonable to start grasping the cup, move to the kitchen door, sense its state and then continue the planning process. In contrast, it usually should be avoided to execute critical actions that cannot be undone until a complete plan is generated. The default strategy of the proposed controller is to execute the whole plan prefix prior to the execution of knowledge acquisition tasks. However, due to the fact this is not always the best strategy it is possible to specify domain specific control rules.

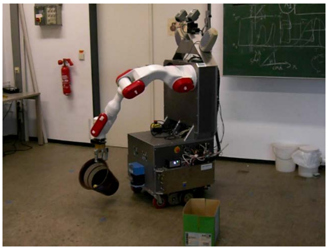
4 Experimental Results

The proposed plan-based control system is implemented and evaluated on the mobile service robot TASER (see Fig. 5(a)) as well as using a set of simulated domains.

4.1 Physical Service Robot

This chapter describes an experiment that was conducted with the service robot TASER. The robot had the task to clean a table. Cleaning a table includes the execution of several typical service robotic tasks including: pick up an object from a table, navigate to a desired goal position, find a garbage can, throw away objects, and pick up a garbage can.

Initially, the robot had no knowledge about dynamic aspects of the environment including: what unknown (i.e., in addition to known objects) dynamic objects exist in the world, the number of objects on the table, the position of the objects on the table, the position of objects on the ground that can obstruct a passage, the location of the garbage can, and the state of the doors.



(a) TASER picks up a garbage can.

Name	aver.	min	max
planning/execution phases	62	56	73
action primitives per phase	5.14	1	27
action primitives per run	313	276	359
percepts	69	58	78
percepts per phase	1.13	1.04	1.20
planning CPU time per action	0.0062 s	–	–
planning CPU time per phase	0.0317 s	0.0005 s	0.1898 s
planning CPU time per run	1.9334 s	1.5396 s	2.2994 s
execution time – action	3.363 s	0.008 s	34.17 s
execution time – run	1056 s	897 s	1259 s

(b) Results for the experiments with the physical service robot TASER.

Fig. 5.

We performed five runs with varying situations (e.g., different position of objects). For all runs, the robot successfully performed the given task. Fig. 5(b) shows additional results. On average, ACogControl divided the overall task of cleaning the table and bringing the garbage out into 62 planning and execution phases, executed 313 action primitives for an experiment run, and planned 5.14 steps (i.e., action primitives) ahead. For the complete execution of the given tasks, 1.9334 seconds CPU time on average is used for planning and reasoning. This is very low compared to the mean execution time, which is 3.363 seconds for an action primitive and 1056 seconds for a complete experiment run. Thus, the ratio of time used for planning and reasoning to the overall execution time is very low at $\frac{1.9334\,\mathrm{s}}{1.9334\,\mathrm{s}+1056\,\mathrm{s}} = 0.0019$.

4.2 ACogSim

Providing an environment for the evaluation of continual planning systems is not a trivial task [2]. We implemented a simulator, namely *ACogSim*, for the environment in order to make it possible to systematically evaluate the whole high-level control architecture—including execution—described in Section 3. The ACogSim simulator works similar to MAPSIM as described in [2]. In contrast to the agent, ACogSim has a complete model of the domain. When the executor executes an action, then the action is sent to ACogSim. ACogSim checks the precondition of actions at runtime prior to the execution and updates its simulation model according to the effect of the actions. In this way ACogSim simulates the execution of actions and guarantees that the executed plans are correct.

The outcome of sensing actions is also simulated by ACogSim. Let D_{Msim} be the (complete) domain model of the ACogSim instance. The result of a sensing action $\texttt{sense}(l, I, C, ks)$ is an additional instance $l\sigma$ of l if such an instance can be derived with respect to D_{Msim}; *impossible* if it can be derived that the existence of an additional instance of l is impossible; or *indeterminable* otherwise.

4.3 Performing Tasks with a Decreasing Amount of Initial Knowledge

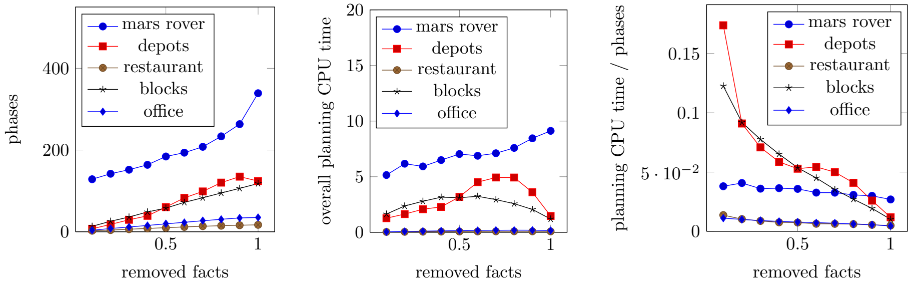
We used ACogSim in order to evaluate the behavior of the overall control system for several domains. The objective of the conducted experiments is to determine the behavior of the system in situations where an agent needs additional information to perform a given task, but sufficient information can in principle be acquired by the agent.

Setup. We used an adapted version of the rover domain with 1756 facts and an instance of the depots domain with 880 facts from IPC planning competition 2002; an instance of an adapted blocks world domain with 2050 facts; and a restaurant (109 facts) and an office domain (88 facts) used to control a mobile service robot.

All domain model instances contain sufficient information to generate a complete plan without the need to acquire additional information. The simulator (ACogSim) is equipped with a complete domain model. In contrast, the agent has only an incomplete domain model where a set of facts has randomly been removed. For each domain the agent always had to perform the same task.

The objective of this experimental setup is to get deeper insights into the performance of the proposed control system. In particular, we are interested in finding an answer to the following questions: Is ACogControl always able to perform the given task? How often switches ACogControl between planning and acting? How much time is necessary for the whole planning and reasoning process? How long is an average planning phase? How does the performance change with a decreasing amount of initial knowledge?

We conducted 10 experiments for all domains with 1000 runs per experiment, except for the last experiment where 1 run was sufficient. Let $\mathfrak{f}_{all}$ be the number of facts in a domain, then $\frac{i}{10}\mathfrak{f}_{all}$ facts were removed in all runs of the ith experiment from the domain model of the agent. Hence, in the last experiment all facts are removed (for each domain) from the agent's domain model.



(a) Average number of planning and execution phases.

(b) Average CPU time of the overall planning and reasoning process.

(c) Average CPU time of a single planning phase.

Fig. 6. System behavior for a decreasing amount of initial knowledge about dynamic aspects of the domain

The experiments where conducted on a 64-bit Intel Core 2 Quad Q9400 with 4 GB memory.

Results. ACogControl was able to correctly perform the given task for all domains and all runs—even in situations where all facts were removed from the domain model of the agent. The average number of necessary planning and execution phases is show in Fig. 6(a). The average number of planning and execution phases increases with a decreasing number of initial information, since the agent needs to stop the planning process and execute knowledge acquisition activities more often. We also expected the overall CPU time of the reasoning and planning process to increase for all domains with a decreasing amount of initial knowledge. However, Fig. 6(b) shows that this is only true for the rover, the office and the restaurant domain. The blocks and the depots domain show a different behavior. For these domains the overall CPU time increases until 60 respectively 80 percent of the facts are removed from the domain model of the agent and then decreases until all facts are removed. The results shown in Fig. 6(c) might give an explanation for this. They show that the average time for a planning phase decreases with a decreasing amount of information that initially is available for the agent. Together with the results shown in Fig. 6(a) these results indicate that the more planning phases are performed the shorter are the individual phases. Thus, the proposed continual planning system, so to speak, partitions the overall planning problem into a set of simpler planning problems. Moreover, the depots and the blocks world domain indicate that the sum of the individual planning phases can be lower even if the number of planning phases is higher as shown by Fig. 6(b).

5 Related Work

Most of the previous approaches that are able to generate plans in partially known environments generate conditional plans—or policies—for all possible contingencies. This includes *conformant*, *contingent* or *probabilistic* planning approaches [13,4]. Several planning approaches that generate conditional plans, including [1,3,6,7], use *runtime variables* for the purpose of representing unknown information. Runtime variables can be used as action parameters and enable the reasoning about unknown future knowledge. Nevertheless, the information represented by runtime variables is limited since the only thing that is known about them is the fact that they have been sensed. Furthermore, planning approaches that generate conditional plans are computationally hard, scale badly in open-ended domains and are only applicable if it is possible to foresee all possible outcomes of a sensing action [4,2].

The most closely related previous work is [2]. The proposed continual planning system also deals with the challenge of generating a plan without initially having sufficient information. In contrast to our work, this approach is based on classical planning systems that do not natively support the representation of incomplete state models and are unable to exploit domain specific control knowledge in the form of HTN methods. Moreover, it is not stated whether the approach can deal with open-ended domains in which it is not only necessary to deal with incomplete information, but also essential to, for example, consider the existence of a priori completely unknown objects or relations between entities of a domain. Furthermore, the approach is based on the assumption that all information about the precondition of a sensing action is a priori available and thus will often (i.e., whenever this information is missing) fail to achieve a given goal in an open-ended domain.

The Golog family of action languages—which are based on the situation calculus [11]—have received much attention in the cognitive robotics community. The problem of performing tasks in open-ended domains is most extensively considered by the IndiGolog language [5], since programs are executed in an on-line manner and thus the language to some degree is applicable to situations where the agent posses only incomplete information about the state of the world. Regrettably, IndiGolog only supports binary sensing actions.

Besides Golog the only other known agent programming language is FLUX [14] which is based on the Fluent Calculus. FLUX is a powerful formalism, but uses a restricted form of conditional planning. As already pointed out, conditional planning is not seen as an adequate approach for the scenarios we are interested in.

6 Discussion and Conclusions

State-of-the-art planning techniques can provide artificial agents to a certain degree with autonomy and robustness. Unfortunately, reasoning about external information and the acquisition of relevant knowledge has not been sufficiently considered in existing planning approaches and is seen as an important direction of further growth [9].

We have proposed a new, continual, HTN-planning-based control system that can reason about possible, relevant, and possibly-acquirable extensions of a domain model. It makes an agent capable of autonomously generating and answering relevant questions. The domain specific information encoded in HTN methods not only helps to prune the search space for classical planning problems but can also be exploited to rule out irrelevant extensions of a domain model.

Planning in open-ended domains is obviously more difficult than planning based on the assumption that all information is available at planning time. Nevertheless, the experimental results indicate that the proposed approach partitions the overall planning problem into a number of simpler planning problems. This effect can make continual planning in open-ended domains sufficiently fast for real world domains. Additionally, it should be considered that the execution of a single action is often much more time intensive for several agents (e.g., robots) than the planning phases of the evaluated domains.

Acknowledgements. This work is founded by the DFG German Research Foundation (grant #1247), International Research Training Group CINACS (Cross-modal Interactions in Natural and Artificial Cognitive Systems).

References

1. Ambros-Ingerson, J.A., Steel, S.: Integrating planning, execution and monitoring. In: AAAI. pp. 83–88 (1988)
2. Brenner, M., Nebel, B.: Continual planning and acting in dynamic multiagent environments. Autonomous Agents and Multi-Agent Systems 19(3), 297–331 (2009)
3. Etzioni, O., Hanks, S., Weld, D.S., Draper, D., Lesh, N., Williamson, M.: An approach to planning with incomplete information. In: KR, pp. 115–125 (1992)
4. Ghallab, M., Nau, D., Traverso, P.: Automated Planning Theory and Practice. Elsevier Science (2004)
5. Giacomo, G.D., Levesque, H.J.: An incremental interpreter for high-level programs with sensing. In: Levesque, H.J., Pirri, F. (eds.) Logical Foundation for Cognitive Agents: Contributions in Honor of Ray Reiter, pp. 86–102. Springer, Berlin (1999)
6. Golden, K.: Leap before you look: Information gathering in the puccini planner. In: AIPS, pp. 70–77 (1998)
7. Knoblock, C.A.: Planning, executing, sensing, and replanning for information gathering. In: IJCAI, pp. 1686–1693 (1995)
8. Littman, M.L., Goldsmith, J., Mundhenk, M.: The computational complexity of probabilistic planning. J. Artif. Intell. Res (JAIR) 9, 1–36 (1998)
9. Nau, D.S.: Current trends in automated planning. AI Magazine 28(4), 43–58 (2007)
10. Nau, D.S., Cao, Y., Lotem, A., Muñoz-Avila, H.: Shop: Simple hierarchical ordered planner. In: IJCAI, pp. 968–975 (1999)
11. Reiter, R.: Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems, illustrated edition edn. The MIT Press (September 2001)
12. Rintanen, J.: Constructing conditional plans by a theorem-prover. J. Artif. Intell. Res (JAIR) 10, 323–352 (1999)
13. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall (2010)
14. Thielscher, M.: FLUX: A logic programming method for reasoning agents. Theory Pract. Log. Program. 5, 533–565 (2005)

Natural Language Interpretation for an Interactive Service Robot in Domestic Domains

Stefan Schiffer, Niklas Hoppe, and Gerhard Lakemeyer

Knowledge-Based Systems Group, RWTH Aachen University, Aachen, Germany
niklas.hoppe@rwth-aachen.de,
{schiffer,gerhard}@cs.rwth-aachen.de

Abstract. In this paper, we propose a flexible system for robust natural language interpretation of spoken commands on a mobile robot in domestic service robotics applications. Existing language processing for instructing a mobile robot is often restricted by using a simple grammar where precisely pre-defined utterances are directly mapped to system calls. These approaches do not regard fallibility of human users and they only allow for binary processing of an utterance; either a command is part of the grammar and hence understood correctly, or it is not part of the grammar and gets rejected. We model the language processing as an interpretation process where the utterance needs to be mapped to the robot's capabilities. We do so by casting the processing as a (decision-theoretic) planning problem on interpretation actions. This allows for a flexible system that can resolve ambiguities and which is also capable of initiating steps to achieve clarification. We show how we evaluated several versions of the system with multiple utterances of different complexity as well as with incomplete and erroneous requests.

Keywords: Natural Language Processing, Interpretation, Decision-theoretic Planning, Domestic Service Robotics, RoboCup@Home.

1 Introduction

In this paper we present a system for flexible command interpretation to facilitate natural human-robot interaction in a domestic service robotics (DSR) domain. We particularly target the *General Purpose Service Robot* test from the RoboCup@Home competition [21], where a robot is confronted with ambiguous and/or faulty user inputs in form of natural spoken language. The main goal of our approach is to provide a system capable of resolving these ambiguities and of interactively achieving user satisfaction in the form of doing the right thing, even in the face of incomplete, ill-formed, or faulty commands.

We model the processing of natural spoken language input as an interpretation process. More precisely, we first analyse the given utterance syntactically by using a grammar. Then, we cast the interpretation as a planning problem where the individual actions available to the planner are to interpret syntactical elements of the utterance. If, in the course of interpreting, ambiguities are detected, the system uses decision-theory to weigh different alternatives. The system is also able to initiate clarification to resolve ambiguities and to handle errors as to arrive at a successful command interpretation

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 39–53, 2013.

eventually. Since our current high-level control already knows about the robot's capabilities (the actions and the parameters that these actions need), we want to tightly connect the interpretation with it. This paper is a revised and extended version of [17].

The remainder of this paper is organised as follows. In the next section we introduce the foundations of our work and we briefly review related work. Then, we go into detail on our approach in Section 3. We present an evaluation in Section 4 before we conclude and present future work in Section 5.

2 Foundations and Related Work

In this section, we introduce the foundations, namely the situation calculus and GOLOG, which our approach is based on. We then briefly review related work.

2.1 Foundations

The high-level control of our domestic service robot uses a logical programming and plan language called READYLOG. It is a dialect of GOLOG which itself is based on the situation calculus.

The Situation Calculus and GOLOG. The situation calculus [14] is a sorted second order logical language with equality that allows for reasoning about actions and their effects. The situation calculus distinguishes three different sorts: *actions*, *situations*, and domain dependent *objects*. The state of the world is characterised by functions and relations with a situation as their last argument. They are called *functional* and *relational fluents*, respectively. The world evolves from an initial situation S_0 only due to primitive actions, e.g., $s' = do(a, s)$ means that the world is in situation s' after performing action a in situation s. Possible world histories are represented as sequences of actions. For each action one has to specify a *precondition axiom* stating under which conditions it is possible to perform the respective action and *effect axioms* formulating how the action changes the world in terms of the specified fluents. An action precondition axiom states when an action can be executed. The effects that actions have on the fluents are described by so-called successor state axioms [16].

GOLOG [13] is a logic-based robot programming and plan language based on the situation calculus. It allows for imperative-style programming but it also offers some non-deterministic constructs. A *Basic Action Theory* (BAT), which is a set of axioms describing properties of the world, axioms for actions and their preconditions and effects as described above, and some foundational axioms, then allows for reasoning about a course of action.

There exist various extensions and dialects to the original GOLOG interpreter, one of which is READYLOG [9]. It integrates several extensions like interleaved concurrency, sensing, exogenous events, and on-line decision-theoretic planning (following [3]) into one framework. In READYLOG programs one can use non-deterministic actions that leave certain decisions open, which then are taken by the controller based on an optimisation theory. The optimisation resembles that of a Markov Decision Process (MDP)

[15]; decision-theoretic planning is initiated with $solve(p, h)$, where p is a GOLOG program and h is the MDP's solution horizon). Two important constructs used in this regard are the non-deterministic choice of actions $(a|b)$ and arguments $(pickBest(v, l, p))$, where v is a variable, l is a list of values to choose from, and p is a GOLOG program. Then each occurrence of v is replaced with the value chosen. For details we refer to [9].

2.2 Related Work

We want to build on the theory of *speech acts* as introduced by Austin [1] and Searle [19]. Based on these works, Cohen and Levesque [5] already investigated a formal theory of rational interaction. We restrict ourselves to command interpretation and do not aim for a full-fledged dialogue system. Nevertheless, we follow their formal theory of interpretation and we carry out our work in the context of the situation calculus.

The use of definite clause grammars for parsing and interpreting natural language has already been shown in [2]. Despite being relatively ad hoc and the fact that the small grammar only covered a constrained subset of English, their system provided a wide spectrum of communication behaviours. However, in contrast to their approach we want to account for incomplete and unclear utterances both by using a larger grammar as well as adding interpretation mechanisms to the system.

[10] developed a system on a robot platform that manages dialogues between human and robot. Similar to our approach, input to the system is processed by task planning. However, queries are limited to questions that can either be answered with yes or no or a decimal value. A more advanced system combining natural language processing and flexible dialogue management is reported on in [4]. User utterances are interpreted as communicative acts having a certain number of parameters. The approach is missing a proper conceptual foundation of objects and actions, though. This makes it hard to adapt it to different platforms or changing sets of robot capabilities.

[11], on the other hand, built a dialogue management system well-founded by making use of a concept hierarchy formalised in Description Logics (DL). Both, the linguistic knowledge as well as the dialogue management are formalised in DL. This is a very generic method for linking lexical semantics with domain pragmatics. However, this comes with the computational burden of integrating description logics and appropriate reasoning mechanisms. We want to stay within our current representational framework, that is, the situation calculus and Golog, and we opt to exploit the capabilities to reduce computational complexity with combining programming and planning.

3 Method and Approach

As mentioned before, we cast the language processing of spoken commands on a domestic service robot as an interpretation process. We decompose this process into the following steps. First, the acoustic utterance of the user is being transformed into text via a speech recognition component which is not part of this paper's contribution. The transcribed utterance is then passed on for syntactic analysis by a grammar. After that, the interpretation starts, possibly resolving ambiguities and generating intermediate responses. If the utterance could be interpreted successfully, it is executed, otherwise it is being rejected. We will now present the individual steps in more detail.

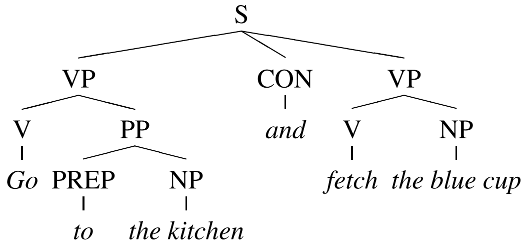


Fig. 1. Syntax tree for the utterance ”Go to the kitchen and fetch the blue cup”

3.1 Syntactical Language Processing

Given the textual form of the user utterance, the first thing we do is a syntactical analysis. This syntactic operation uses a grammar. Since the entirety of the English language is not context-free as revealed by [20] and the targeted application domain allows for a reasonable restriction, we confine ourselves to directives. Directives are utterances that express some kind of request. Following Ervin-Tripp [8] there are six types of directives:

1. Need statements, e.g., “I need the blue cup.”
2. Imperatives, e.g., “Bring me the blue cup!”
3. Imbedded imperatives, e.g., “Could you bring me the blue cup?”
4. Permission directives, e.g., “May I please have the blue cup?”
5. Question directives, e.g., “Have you got some chewing gum?”
6. Hints, e.g., “I have run out of chewing gum.”

Ervin-Tripp characterises question directives and hints as being hard to identify as directives even for humans. Moreover, permission directives are mostly used only when the questioner is taking a subordinate role, which will not be the case of a human instructing a robot. That is why we restrict ourselves to a system that can handle need statements, imperatives and imbedded imperatives only.

A Grammar for English Directives. For any of these directives what we need to make the robot understand the user's command is to distill the *essence* of the utterance. To eventually arrive at this, we first perform a purely syntactic processing of the utterance. An analysis of several syntax trees of such utterances revealed structural similarities that we intend to capture with a grammar. An example for a syntax tree is given in Figure 1.

Using common linguistic concepts, the main structure elements are verb (V), auxiliary verb (AUX), verb phrase (VP), noun phrase (NP), conjunction (CON), preposition (PREP), and prepositional phrase (PP). A verb phrase consists of a verb and a noun phrase, a noun phrase is a noun, possibly having a determiner in front of it. A prepositional phrase is a preposition and a noun phrase. We further introduce a structure element *object phrase* which is a noun phrase, a prepositional phrase, or concatenations of the two. Multiple verb phrases can be connected with a conjunction. What is more, commands to the robots may be prefixed with a salutation. Also, for reasons of

```
                  s —> salutation utterance | utterance
%
          utterance —> needstatement | imperative | imbedded_imperative
%
       needstatement —> np vp | needphrase vp
          imperative —> vp
imbedded_imperative —> aux np vp
          needphrase —> "i" prompt "you to"
% verb phrase
                  vp —> vp' | vp' conjunction vp
                 vp' —> verb | verb obp | courtesy vp'
% object phrase
                 obp —> np | pp | np obp | pp obp
% noun phrase
                  np —> noun | pronoun | determiner noun
% propositional phrase
                  pp —> prep np
```

Fig. 2. Base grammar in EBNF

politeness, the user can express courtesy by saying "please". Putting all this together, we arrive at a base grammar that can be expressed in Extended Backus-Naur Form (EBNF) [18] as shown in Figure 2.

In addition to the base grammar we need a base lexicon that provides us with the vocabulary for elements such as prepositions, auxiliary verbs, courtesies, conjunctions, determiners, and pronouns. To generate a system that is functional in a specific setting, we further need a lexicon containing all verbs for the capabilities of the robot as well as all the objects referring to known entities in the world. This depends on the particular application, though. That is why we couple this to the domain specification discussed later. The base grammar, the base lexicon, and the domain specific lexicon then yield the final grammar that is used for syntactical processing.

Since we are only interested in the core information, the most relevant parts of the utterance are verbs, objects, prepositions, and determiners. We can drop auxiliary verbs, filler words, courtesies, and alike without losing any relevant information. Doing so, we finally arrive at an internal representation of the utterance in a prefix notation depicted below, that we use for further processing.

```
[and, [[Verb, [objects, [[Preposition, [Determiner,Object]],...]] ]], ...]
```

The list notation contains the keyword *`and`* to concatenate multiple verb phrases and it uses the keyword *`objects`* to group the object phrase. If an utterance is missing, information we fill this with *`nil`* as a placeholder.

3.2 Planning Interpretations

After syntactic pre-processing of an utterance into the internal representation, the system uses decision-theoretic planning to arrive at the most likely interpretation of the utterance, given the robot's capabilities. The interpretation is supposed to match the request with one of the abilities of the robot (called a skill) and to correctly allocate the parameters that this skill requires.

In order to do that, we need to identify the skill that is being addressed first. We are going about this from the verb which has been extracted in the syntactical processing, possibly leaving ambiguities on which skill is referred to by the verb. Secondly, the objects mentioned in the utterance need to be mapped to entities in the world that the robot knows about. Lastly, a skill typically has parameters, and the verb extracted from the utterance has (multiple) objects associated to it. Hence, we need to decide which object should be assigned to which parameter. To make things worse, it might very well be the case that we have either too many or too few objects in the utterance for a certain skill.

We cast understanding the command as a process where the single steps are interpretation actions, that is, interpreting the single elements of the utterance. At this point READYLOG and its ability to perform decision-theoretic planning comes into play. The overall interpretation can be modelled as a planning problem. The system can choose different actions (or actions with different parameters) at each stage. Since we want to achieve an optimal interpretation, we make use of decision-theoretic planning. That is to say, given an optimisation theory, we try to find a plan, i.e. a sequence of actions, which maximises the expected reward.

Domain Specification. During the interpretation process we need to access the robot's background knowledge. We organise this knowledge to capture generic properties and to make individual parts available to (only) those components which need them. Three types of information are distinguished: *linguistic*, *interpretation*, and *system*. The linguistic information contains everything that has to do with natural language while interpretation information is used during the interpretation process and system information features things like the specific system calls for a certain skill. The combination of these three types is then what makes the connection from natural language to robot abilities. We use ideas from [12] to structure our knowledge within our situation calculus-based representation.

In an ontology, for every *Skill* we store a *Name* as an internal identifier that is being assigned to a particular skill during the interpretation. A skill further has a *Command* which is the denotation of the corresponding system call of that skill. *Synonyms* is a list of possible verbs in natural language that may refer to that skill. *Parameters* is a list of objects that refer to the arguments of the skill, where *Name* again is a reference used in the interpretation process, *Attributes* is a list of properties such as whether the parameter is numerical of string data. *Significance* indicates whether the parameter is optional or required, and *Preposition* is a (possibly empty) list of prepositions that go with the parameter. For the information on entities in the world (e.g. locations and objects) we use a structure *Object* which again has a *Name* as an internal identifier used during the interpretation. *Attributes* is a list of properties such as whether the object "is a location" or if it "is portable". *Synonyms* is a list of possible nouns that may refer to the object and *ID* is a system related identifier that uniquely refers to a particular object.

Basic Action Theory. Now that we have put down the domain knowledge on skills and objects, we still need to formalise the basic action theory for our interpretation system. We therefore define three actions, namely $interpret\_action$, $interpret\_object$, and $assign\_argument$. For all three we need to state precondition axioms and successor state axioms. We further need several fluents that describe the properties of the interpretation domain we operate in. Let's take a look at those fluents first. We use the fluents $spoken\_verb(s)$ and $spoken\_objects(s)$ to store the verb and the list of objects extracted in the syntactic processing. Further, we use the fluents $assumed\_action(s)$ and $assumed\_objects(s)$ to store the skill and the list of objects that we assume to be addressed by the user, respectively. Both these fluents are nil in the initial situation S_0 since no interpretation has taken place so far. The fluent $assumed\_arguments(s)$ contains a list of pairings between parameters and entities. Finally, $finished(s)$ indicates whether the interpretation process is finished.

Let us now turn to the three interpretation actions. The precondition axiom for $interpret\_action$ states that $interpret\_action(k)$ is only possible if we are not done with interpreting yet and the word k actually is a synonym of the verb spoken. Similarly, $interpret\_object(e)$ is possible for an entity e only if we are not finished and the object (from $spoken\_object(s)$) is a synonym appearing for e. Finally, the precondition axiom for $assign\_argument$ for an entity e and parameter p checks whether the interpretation process is not finished and there is no entity assigned to the parameter yet. Further, p needs to be a parameter of the assumed skill and we either have no preposition for the object or the preposition we have matches the preposition associated with the parameter. Lastly, the attributes associated to parameter p need to be a subset of the attributes for the entity. To allow for aborting the interpretation process we additionally introduce an action $reject$ which is always possible. We omit the formal definitions here for space reasons.

After detailing the preconditions of actions, we now lay out how these actions change the fluents introduced above. The fluents *spoken_verb* and *spoken_objects* contain the essence of the utterance to be interpreted. The effect of the action $interpret\_action(k)$ is to reset the fluent *spoken_verb* to nil and to set the fluent *assumed_action* to the assumed skill k. The action $interpret\_object(e)$ iteratively removes the first object (in a list of multiple objects) from the fluent *spoken_objects* and adds it to the fluent *assumed_objects* along with its preposition (if available). The action *assign_argument*(p) removes the object from the fluent *assumed_objects* and it adds the pair (p, e) for parameter p and entity e to the fluent $assumed\_arguments$. Finally, the fluent *finished* is set to $true$ if either the action was $interpret\_action$ and there are no more objects to process (i.e. *spoken_objects* is empty) or the action was $assign\_argument$ and there are no more objects to assign (i.e. *assumed_objects* is empty). It is also set to $true$ by the action $reject$.

Programs. Using the basic action theory described above, the overall interpretation process can now be realised with READYLOG programs as follows. In case of multiple verb phrases we process each separately. For each verb phrase, we first interpret the verb. Then, we interpret the objects before we assign them to the parameters of the skill determined in the first step. The procedures to do so are

```
proc $interpret\_verbphrase$
    $solve($ {
      ( pickBest( $var$, $AllActions$,
            $interpret\_action(var)$ )
      | $reject$ )
      while $\neg finished$ do
          $interpret\_objectphrase$ endwhile
    }, $horizon$, $reward\_function$ )
endproc
```

with

```
proc $interpret\_objectphrase$
  (pickBest( $var$, $AllEntities$,
                  $interpret\_object(var)$ )
   | $reject$)
  if $finished$ then $nil$
  else
    (pickBest( $var$, $AllParams$,
                  $assign\_argument(var)$ )
     | $reject$)
  endif
endproc
```

where *AllActions*, *AllEntities*, and *AllParams* are sets of all skills of the robot, all entities known to the robot, and all parameters of a skill in the robot's domain specification, respectively. We consider more intelligent selection methods than taking all items available in the evaluation. The *solve*-statement initiates decision-theoretic planning, where **pickBest**$(var, VarSet, prog)$ is a non-deterministic construct that evaluates the program *prog* with every possibility for *var* in *VarSet* using the underlying optimisation theory given mainly by the reward function, which rates the quality of resulting situations. To design an appropriate reward function situations that represent better interpretations need to be given a higher reward than those with not so good interpretation. A possible reward function could be to give a reward of 10 if the assumed action is not *nil* and one could further add the difference between the number of assigned arguments and the total number of parameters required by the selected skill. Doing so results in situations with proper parameter assignment being given higher reward than those with fewer matches. If two possible interpretation have the same reward, one can either ask the user which action to take or simply pick one of them at random.

Example. Consider the utterance "Move to the kitchen." After syntactical processing we have the internal representation `[`*`and`*`, [[move, [`*`objects`*`, [[to, [the, kitchen]] ]]] ] ]`. Using the program given above and a small basic action theory as introduced before, one of the skills available to the robot that has *go* as a synonym may be *goto* which is stored in *assumed_action* by the action *interpret_action*. Then, $interpret\_object(kitchen)$ will assume *kitchen* as the object (along with the preposition *to*). However, it could also interpret "move" as bringing some object somewhere which leads to a lower reward, because a parameter slot remains unassigned. Trying to assign arguments for the skill *goto* may succeed since *kitchen* is an entity that has the *Location* attribute as would naturally be required for the target location parameter of a *goto* skill. Comparing the rewards for the different courses of interpretation the system will pick the interpretation with the highest reward, which is executing the $goto(kitchen)$ skill.

3.3 Clarification and Response

Things might not always go as smoothly as in our example above. To provide a system that has capabilities beyond a pure interface to translate utterances to system calls we therefore include means for clarification if the utterance is missing information.

If the verb is missing, our grammar from the syntactical processing will already fail to capture the utterance. Hence, we only consider missing objects for clarification in the following. We propose to model clarification as an iterative process where the user is questioned for each missing object. To generate the appropriate questions to the user we make use of the information that has been extracted from the utterance already and of the information stored in the ontology. Assuming that we know about the skill that is being addressed we can look up the parameters required. Using a template that repeats the user's request as far as it has been interpreted we can then pose an accurate question and offer possible entities for the missing objects.

Consider that the user said "*Go!*" missing the required target location. So the target location is what we want to enquire about. This can be achieved with using a generic template as follows:

$$\text{“}you\ want\ me\ to\ [assumed\_action]\ [assumed\_arguments].$$
$$[preposition]\ which\ [attribute]\ ?\ \ [list\ of\ entities]\text{”}$$

where $[preposition]$ is the preposition associated to the parameter in question and $[attribute]$ is one of the attributes associated to the parameter. Only including one of the parameter's attributes seems incomplete, but suits the application, since it still leads to linguistically flawless responses. Including $[assumed\_arguments]$ in the response indicates what the system has already managed to interpret and additionally reminds the user of his original request. The system would respond to the utterance "*Go!*" from above with "*You want me to go. To which location? kitchen or bath?*", which is exactly what we want.

To avoid annoying the user we put a limit on the number of entities to propose to the user. If the number of available entities exceeds, say, three we omit it from the question. Moreover, to improve on the response we add what we call "unspecific placeholders" to the domain ontology. So for locations we might add "*somewhere*" and for portable thing we might add "*something*" which are then used in the response at the position of a missing object.

There might be cases where information is not missing but instead is either wrong or the skills available to the robot do not allow for execution. Our system should provide information on rejecting faulty or non-executable requests. Depending on the type of error, we propose the following templates for explanation.

1. "*I cannot [spoken_verb].*" if the verb could not be matched with any skill, i.e. $spoken\_verb \neq nil$.
2. "*I do not know what [next spoken_object] is.*" if the object could not be matched with any entity known to the robot, i.e. $spoken\_objects \neq nil$.
3. "*I cannot [assumed_action] [preposition] [next assumed_object].*" if the object could not be assigned to a parameter of the skill that is being addressed, i.e. $assumed\_objects \neq nil$.

Note that *[next some_list]* retrieves the next element from *some_list*. Also note that the fluent values we mentioned above are sound given our basic action theory since the action *reject* sets the fluent *finished* to true and leaves the other fluents' values as they were when the utterance was rejected.

4 Experimental Evaluation

To investigate the performance of our system we evaluate it along two dimensions, namely understanding and responsiveness.

Table 1. Survey results by sentence type

type	absolute frequency	relative frequency
imperatives	114	87%
imbedded imperatives	6	5%
need-statements	2	2%
hints	4	3%
wh-questions	3	2%
others	3	2%

4.1 Understanding

The aim of our approach was to provide a system that is able to react to as many commands for a domestic service robot given in natural language as possible. With the generic grammar for English directives our approach is able to handle more utterances than previous approaches based on finite state grammars such as [7]. To evaluate how far off we are from an ideal natural language interface we conducted a user survey. The survey was carried out on-line with a small group of (about 15) predominantly tech-savvy students. A short description of the robot's capabilities was given and participants were asked to provide us with sample requests for our system. Participants took the survey without any assistance, except the task description.

We received a total of 132 submissions. Firstly, we are interested in the general structure of the answers to see whether our grammar is appropriate. Therefore, Table 1 shows the submissions itemised by sentence type.

Syntactically speaking, the grammar can cover imperatives, imbedded imperatives and need-statements, which make for 92.37% of the survey results. However, some of these utterances do not possess the verb-object-structure we assumed in our system. For example, "Make me a coffee the way I like it" contained an adverbial ("the way I like it") which we did not account for neither in the grammar nor in the interpretation process. It is technically possible to treat adverbials as entities and thus incorporate such utterances. A better founded approach, however, would be to introduce the concept of adverbials to our system as a special case of objects that modify the mode of a skill. We leave this for future work, though. Still, 77.01% of the survey entries provide the assumed modular verb-object-structure and can therefore be processed by our system successfully.

To test the resilience against erroneous utterances we tested the system's response to the set of utterances given in Table 2. In case that an object is missing that is required as a parameter by a skill (as in E1) the system will inquire for clarification by offering possible entities. To be able to handle unspecific objects we included those in our grammar and we treat them just like missing objects and initiate a clarification procedure. Preposition help in assigning objects to parameter slots of a skill. With only one parameter as in the utterance E3 we do not require the preposition in order to come to a successful termination of our interpretation process. With multiple parameters that are identical in all their attributes we would need additional information though. We do not make use of prepositions to resolve these kinds of confusions, yet. An utterance

Table 2. Types of erroneous utterances

id	example utterance	problem
E1	"fetch"	object missing
E2	"go somewhere"	unspecific object
E3	"go the kitchen"	missing preposition
E4	"collect the bath room"	nonsense
E5	"smurf"	unknown word
E6	"the cup i need" or "the cup"	ill-formed syntax

can be nonsense when the objects do not match the attributes required for a parameter as specified in our ontology. In cases such as the one in E4 the system rejects the utterance since "the bath room" is not a "portable object" as required for the "collect" skill. This is then also mentioned in an explanation given to the user. Words that do not occur in our lexicon can not be processed. Hence, the system will fail when confronted with unknown words. When the system is confronted with ill-formed syntax, it fails at the syntactical processing stage. This is because the grammar cannot handle utterances with unknown constructions.

4.2 Responsiveness

To evaluate the performance of our system in terms of speed, we evaluated the system using the following domain. The example agent has four different skills: getting lost (no parameter), going somewhere (1 parameter), moving an object to some location (2 parameters) and moving an object from some location to some location (3 parameters). Additionally, our domain contains different entities with appropriate attributes: a kitchen (location), a bath (location), a coffee cup (portable object) and a football trophy (decoration). Some of the synonyms for skills and entities are ambiguous, namely (1) "go" may refer to "get lost" as well as to "go somewhere", (2) "move" may refer to "get lost", "go somewhere", "move something somewhere" or "move something from somewhere to somewhere", and (3) "cup" may refer to the coffee cup as well as to the football trophy.

We tested four different versions of the system with different requests involving various degrees of complexity using the following utterances:

(i) "scram"
(ii) "go to the kitchen"
(iii) "could you please move the cup to the kitchen"
(iv) "go to the kitchen and move the cup to the bath room"
(v) "i need you to move the cup from the bath room to the kitchen"

Utterance (i) is a very simple request. It addresses a skill with no parameters and the used synonym "scram" is unambiguous. The skill addressed in utterance (ii) involves one parameter and the used synonym "go" is ambiguous. Utterance (iii) involves a skill with two parameters and the synonym "move" is also ambiguous. Utterance (iv) is the combination of utterances (ii) and (iii) linked with an "and". The skill requested in utterance (v) has three parameters and the synonym "move" is again ambiguous.

The depth of the search tree spanned in the planning process depends on the number of objects. For example, the depth of the search tree for utterance (i) is exactly 1

Table 3. Response times in different test scenarios

	i	**ii**	**iii**	**iv**	**v**
base	0.08 s	0.28 s	2.37 s	2.67 s	9.06 s
action pre-select	0.08 s	0.24 s	2.10 s	2.29 s	7.15 s
entity pre-select	0.06 s	0.19 s	2.01 s	2.16 s	7.41 s
parameter pre-select.	0.09 s	0.19 s	1.06 s	1.20 s	4.05 s
action + entity	0.05 s	0.16 s	1.70 s	1.85 s	6.07 s
entity + parameter	0.05 s	0.13 s	0.99 s	1.10 s	3.75 s
action + parameter	0.09 s	0.13 s	0.71 s	0.83 s	2.52 s
full combination	0.07 s	0.10 s	0.68 s	0.76 s	2.35 s

while the depth of the search tree for utterance (v) is 7. Note that utterance (iv) involves two distinct search trees, since it contains two independent verb phrases which are interpreted separately.

The five utterances were tested with the following versions of the system. First, we used the base system as described in Section 3, it does not include any explicit performance improvements speed-wise. The first row of Table 3 shows the performance of the base system.

Improvements. Second, we considered systems incorporating different pre-selection methods. For each interpretation step (interpreting action, entity and parameter), we can pre-select the candidates that may be considered by the appropriate interpretation action. This can lead to considerably lower branching factors.

The pre-selection process for $interpret\_action$ involves two criteria: synonym and parameter count. This means that candidates are eliminated from the list if the spoken verb is not one of the candidates' synonyms or if the number of parameters the candidate provides is lower than the number of spoken objects. This is due to the fact that we want every spoken object to be assigned to a parameter slot, so we only have to consider skills that provide a sufficient amount of parameter slots. If we would also consider skills with fewer parameters, we would have to drop parts of the user's utterance. One could argue that reducing the set of available skills is a restriction from a theoretical point of view. However, ignoring elements that where uttered could easily frustrate the user. Hence, the restriction only has little practical relevance. The second row of Table 3 illustrates the performance of the base system plus *action pre-selection*.

Entities are pre-selected just by checking whether the spoken object is one of the entity's synonyms. The third row of Table 3 shows the response times including the base system plus *entity pre-selection*.

Pre-selecting parameters involves checking the attributes and the preposition of the corresponding candidate. Hence, the attributes of the parameter slot have to be a subset of the entities attributes, and if a preposition was provided along with the spoken object or entity, respectively, then it has to match the preposition required by the parameter. The fourth row of Table 3 lists response times of the base system plus *parameter pre-selection*. Rows five, six and seven illustrate the performance of different pairs of the

Table 4. Response times (in seconds) depending on the two types of difficulty

# of	**tree**	**#actions/#entities**			
obj.	**depth**	**1/1**	**1/5**	**5/1**	**5/5**
1	3	0.15	0.32	0.48	1.27
2	5	0.47	0.96	1.61	3.50
3	7	2.54	4.83	7.40	13.92
4	9	18.77	34.00	39.72	68.19
5	11	153.40	267.55	154.97	276.20

three pre-selection methods. The last row shows the performance of the system including all three enhancements. As we can see, the full combination yields an improvement except for utterance i where the difference is negligible. The relative improvement of the enhancements increases with the complexity of the utterances. That is to say, the more complex the utterance, the more the speed-ups pay off.

Altogether, the complexity of the search tree is affected by the different branching factors at each level, and the depth which depends on the number of spoken objects. The branching factor at the first level depends on the number of actions that have the spoken verb as a synonym. The branching factor at the second level depends on the number of entities that have the spoken object as a synonym. At the third level the branching factor depends on the number of parameters of the respective skill. We further evaluated our optimised system by varying the two complexity factors independently.

Along the rows of Table 4 we varied the number of spoken objects. Along the columns we varied the number of actions that have the spoken verb as a synonym and the number of entities that have the spoken object as a synonym. The number of parameters of the appropriate skill are not varied, since this number already depends on the amount of spoken objects. In this test scenario the parameters of a skill became distinguishable for the system by providing distinct prepositions for each parameter. Different entities became distinguishable through their attributes and the skills were distinguishable by the number of parameters. So we had five skills with 1, 2, 3, 4 and 5 parameters, respectively.

Table 4 shows that the number of spoken objects has a greater influence on the computation time than has ambiguity. This is indicated by the last two rows which only contain measurements greater than 10 seconds. That is unacceptable for fluent human-robot interaction. We can also observe that action pre-selection performs very well in this test scenario. All tests in the last row address a skill with five parameters. In this test scenario there was no other skill involving five or more parameters. As a consequence, the action pre-selection can rule out the other four skill candidates which implies nothing less than reducing the branching factor of the top node from 5 to 1 and thus reducing the computation time by a factor of approximately 5. This also results in comparable computation times for the combinations 1/1 (153.40 sec) and 5/1 (154.97 sec) as well as 1/5 (267.55 sec) and 5/5 (276.20 sec).

Finally, we analysed whether the lexicon size poses a computational problem. Therefore, we simply added 50,000 nouns to the lexicon and used the full combination test setup from Table 3. Now, Table 5 indicates that the additional computational effort to process the utterances with a large lexicon plays no significant role.

Table 5. Response times with different lexicons

	small lexicon	**large lexicon**
utt. i	0.07 sec	0.08 sec
utt. ii	0.10 sec	0.14 sec
utt. iii	0.68 sec	0.90 sec
utt. iv	0.76 sec	1.15 sec
utt. v	2.35 sec	2.51 sec

4.3 Discussion

An important point towards successful human-robot interaction with respect to the user's patience is the system's reaction time. The average human attention span (for focused attention, i.e. the short-term response to a stimulus) is considered to be approximately eight seconds [6]. Therefore, the time we require to process the utterance of a user and react in some way must not exceed 8 seconds. Suitable reactions are the execution of a request, rejection, or to start a clarification process.

Hence, the question whether computation times are reasonable is in fact the question whether the computation times exceed eight seconds. Nonetheless, the answer is not as easy as the question. The optimised system performs well in a realistic test scenario as shown by the last row of Table 3. In turn, complex test scenarios can lead to serious problems as Table 4 indicated. However, we saw that ambiguity is a smaller problem than the length of an utterance[1]. Skills that have more than three parameters are rare in the field of mobile service robots. In fact, the skills with four or five parameters we used in the tests of Table 4 needed to be created artificially in lack of realistic examples.

5 Conclusions and Future Work

We presented a system for interpreting commands issued to a domestic service robot using decision-theoretic planning. The proposed system allows for a flexible matching of utterances and robot capabilities and is able to handle faulty or incomplete commands by using clarification. It is also able to provide explanations in case the user's request cannot be executed and is rejected. The system covers a broader set of possible requests than existing systems with small and fixed grammars. Also, it performs fast enough to prevent annoying the user or loosing his or her attention.

Our next step is to deploy the system in a RoboCup@Home competition to test its applicability in a real setup. A possible extension of the approach could be to include a list of the n most probable interpretations and to verify with the user on which of these should be executed. Moreover, properly integrating the use of adverbials as qualifiers for nouns both in the grammar and the interpretation process would further improve the system's capabilities.

[1] By the length of an utterance, we mean the number of spoken objects.

References

1. Austin, J.L.: How to Do Things with Words, 2 edn. Harvard University Press (1975)
2. Beetz, M., Arbuckle, T., Belker, T., Cremers, A.B., Schulz, D.: Integrated plan-based control of autonomous robots in human environments. IEEE Intelligent Systems 16(5), 56–65 (2001)
3. Boutilier, C., Reiter, R., Soutchanski, M., Thrun, S.: Decision-theoretic, high-level agent programming in the situation calculus. In: Proc. of the 17th Nat'l Conf. on Artificial Intelligence (AAAI 2000), pp. 355–362. AAAI Press/The MIT Press (2000)
4. Clodic, A., Alami, R., Montreuil, V., Li, S., Wrede, B., Swadzba, A.: A study of interaction between dialog and decision for human-robot collaborative task achievement. In: Proc. of the International Symposium on Robot and Human Interactive Communication (RO-MAN 2007), August 26-29, pp. 913–918 (2007)
5. Cohen, P.R., Levesque, H.J.: Speech acts and rationality. In: Proc. of the 23rd Annual Meeting on Association for Computational Linguistics, pp. 49–60 (1985)
6. Cornish, D., Dukette, D.: The Essential 20: Twenty Components of an Excellent Health Care Team. RoseDog Books (2009)
7. Doostdar, M., Schiffer, S., Lakemeyer, G.: A Robust Speech Recognition System for Service-Robotics Applications. In: Iocchi, L., Matsubara, H., Weitzenfeld, A., Zhou, C. (eds.) RoboCup 2008. LNCS (LNAI), vol. 5399, pp. 1–12. Springer, Heidelberg (2009)
8. Ervin-Tripp, S.: Is Sybil there? The structure of some American English directives. Language in Society 5(01), 25–66 (1976)
9. Ferrein, A., Lakemeyer, G.: Logic-based robot control in highly dynamic domains. Robotics and Autonomous Systems 56(11), 980–991 (2008), Special Issue on Semantic Knowledge in Robotics
10. Fong, T., Thorpe, C., Baur, C.: Collaboration, dialogue, human-robot interaction. In: Robotics Research. Springer Tracts in Advanced Robotics, vol. 6, pp. 255–266. Springer (2003)
11. Görz, G., Ludwig, B.: Speech Dialogue Systems - A Pragmatics-Guided Approach to Rational Interaction. KI–Künstliche Intelligenz 10(3), 5–10 (2005)
12. Gu, Y., Soutchanski, M.: Reasoning about large taxonomies of actions. In: Proc. of the 23rd Nat'l Conf. on Artificial Intelligence, pp. 931–937. AAAI Press (2008)
13. Levesque, H.J., Reiter, R., Lespérance, Y., Lin, F., Scherl, R.B.: Golog: A logic programming language for dynamic domains. J. Logic Program 31(1-3), 59–84 (1997)
14. McCarthy, J.: Situations, Actions, and Causal Laws. Technical Report Memo 2, AI Lab, Stanford University, California, USA (July 3, 1963)
15. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc. (1994)
16. Reiter, R.: Knowledge in Action. Logical Foundations for Specifying and Implementing Dynamical Systems. MIT Press (2001)
17. Schiffer, S., Hoppe, N., Lakemeyer, G.: Flexible command interpretation on an interactive domestic service robot. In: Proc. of the 4th International Conference on Agents and Artificial Intelligence (ICAART), pp. 26–35. SciTePress (2012)
18. Scowen, R.: Extended BNF – generic base standards. In: Proc. of the Software Engineering Standards Symposium, August 30-September 03, pp. 25–34 (1993)
19. Searle, J.R.: Speech Acts: An Essay in the Philosophy of Language. Cambridge University Press, Cambridge (1969)
20. Shieber, S.: Evidence against the context-freeness of natural language. Linguistics and Philosophy 8(3), 333–343 (1985)
21. Wisspeintner, T., van der Zant, T., Iocchi, L., Schiffer, S.: Robocup@home: Scientific Competition and Benchmarking for Domestic Service Robots. Interaction Studies. Special Issue on Robots in the Wild 10(3), 392–426 (2009)

A Fuzzy Logic Modeling Approach to Assess the Speed Limit Suitability in Urban Street Networks

Yaser E. Hawas and Md. Bayzid Khan

Roadway, Transportation and Traffic Safety Research Center, UAE University,
Al Ain, United Arab Emirates
{y.hawas@uaeu.ac.ae}

Abstract. This paper discusses the development of fuzzy logic model for estimating the 85th percentile speed of urban roads. Spot speed survey was conducted on four randomly selected urban road segments for a typical weekday and a weekend. The considered road segment attribute data are length of the road segment, number of access points/intersecting links, number of pedestrian crossings, number of lanes, hourly traffic volume, hourly pedestrian volume and current posted speed limits of the selected roads. Such attribute data were collected and used as input variables in the model. Two models for weekday and weekend were developed based on the field survey data. Both models were calibrated using the neuro-fuzzy technique for optimizing the fuzzy logic model (FLM) parameters. Analyses of estimated results show that the FLM can estimate the 85th percentile speed to a reasonable level.

Keywords: 85th Percentile Speed, Posted Speed Limit, Fuzzy Logic, Neuro-fuzzy Training.

1 Introduction

Determining a safer posted speed limit for any given roads/links is one of the major challenges for the researchers and professionals all around the world. Many studies tried to identify the safer speed limit for a road [1-3]. Setting a speed limit is a multi-criteria task. Many road and roadside factors such as the road alignment, section length, traffic volume, pedestrian volume, current speed limit, number of lanes, weather condition, time of the day, law enforcement, purpose and length of the trip, vehicles' characteristics are to be incorporated. [4-5]. Setting the speed limits also requires understanding the drivers' characteristics and their driving pattern. As such, most of the studies suggested the 85th percentile of the operating speed to be set as the posted speed limit [6].

Studies showed that the chances of involving in a crash is least at 85th the percentile traffic speed [1], [7].

Developing a model to estimate the 85th percentile speed by incorporating all the factors is quite challenging. The individual driver's operating speed depends on individual driver's perception about all of the above mentioned factors. For a given road characteristics, every driver may choose different operating speed. Therefore, it is very important to develop a method to estimate the 85th percentile speed which will also address such uncertain choice behavior.

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 54–68, 2013.

Many studies were conducted to determine the factors that influence the choice of the operating speed. Poe, Tarris and Mason [8] showed that access points, land-use characteristics and traffic engineering features have influences on vehicle speed on low speed urban streets. Haglund and Aberg [9] showed that the posted speed limit has influence on drivers' speed choice behavior. Fitzpatrick, Carlson, Brewer and Wooldridge [10] evaluated the influence of geometric, roadside and traffic control devices on drivers' speed on four-lane suburban arterials and found that posted speed limit was the most significant variable for both curve and straight sections. Wang [11] demonstrated that the number of lanes, sidewalks, pedestrian movements, and access density have significant influences on the drivers' behavior of choosing operating speed. Fildes, Fletcher and Corrigan [12] and Fildes, Leening and Corrigan [13] found that the road width and the number of lanes have the greatest influence on speed choice. Tignor and Warren [14] showed that the number of access points and the nearby commercial development have the greatest influences on the vehicle speeds. Most of these studies used different model approaches range from simple linear regression models to complex curvilinear regression equations [8], [11], [15]. Most of the existing models attempt to quantify the operating speed based on physical conditions such as road geometric design, roadside development and traffic control devices. All of these models used 85th percentile speed as a representative measures for operating speed.

No studies on the use of FLM to estimate the 85th percentile speed have been found. The FLM approach has the premise to tackle the imprecise, vague and uncertain relationship between the inputs and outputs. The proposed system can be regarded as an expert system or a knowledge base. It is critically important that the design of such system should account for the imprecise, incomplete or not totally reliable information [16]. The key feature of the FLM is the suitability to incorporate intuition, heuristic and human reasoning [17] and such technique is useful for problems that entail probabilistic or stochastic uncertainty (human behavior modeling), or problems that cannot be easily represented by mathematical modeling because of the complexity of the process [18]. Fuzzy set theory provides a strict mathematical framework in which vague conceptual phenomena can be precisely and rigorously studied [19]. The word imprecise or vague does not mean the lack of knowledge of data; rather it indicates the sense of vagueness of the value of parameters.

The objective of this paper is to develop a fuzzy logic based approach to estimate the 85th percentile speed for different urban road segments based on road segments attribute data for weekday and weekend. In doing so, four urban road segments (one local and three arterial roads) of Al Ain city of United Arab Emirates have been selected randomly (termed as 'Site 1' to 'Site 4'). Only four road segments were selected because of limited time and resources for conducting the study. The authors do recognize that the limited data collection cannot be used to make general conclusions on the validity of the devised FLM for a general network. We emphasize here that the main contribution of this study is the introduction of the concepts and the procedure to develop the FLM that can be generalized to any network given that adequate data collection on a representative sample size is fulfilled.

This paper is divided into five sections. The second section provides a brief over view on data collection methodology. In third section, the structure of the proposed FLM is discussed in brief. The inference engine and fuzzy operators, and neuro-fuzzy training procedure are also discussed. The fourth section discusses the FLM validation and analysis of results. Concluding remarks on the use of the FLM for estimating the 85th percentile speed to set the speed limit are provided in the last section.

2 Data Collection

Spot speed survey were conducted on selected four sites for five different time periods of the day, for a typical weekday and weekend and for both directions. The five time periods include both peak (AM, MD, PM) and off-peak periods (15 minutes within each time period). Only passenger vehicles (excluding trucks and busses) were selected randomly for the survey, keeping in mind that a minimum of 50 vehicles should be observed for spot speed study on each selected road segments [20]. The 85th percentile speed of the spot speed data was calculated for 40 different cases (4 sites*2directions*5 time periods) for two days (one typical weekday and one weekend).

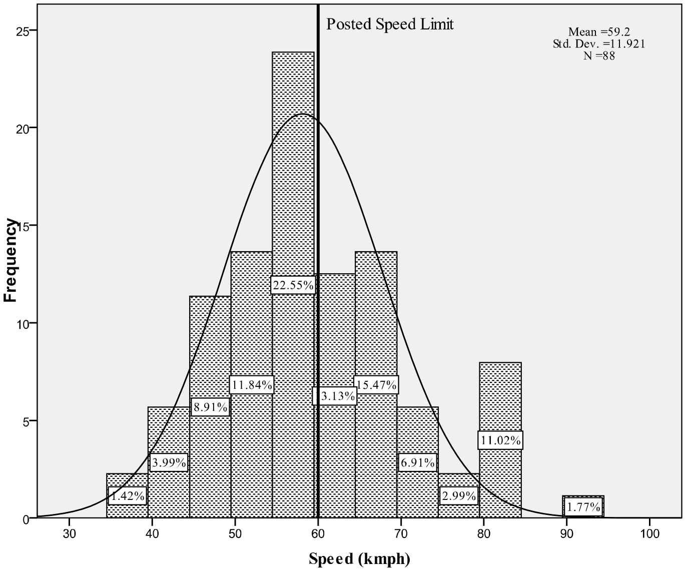


Fig. 1. Observed spot speed distribution at Site 1 direction 1 for weekday (6:30 -7:30 AM)

The detailed road attribute data including the length of the road segment, number of access points/intersecting links, number of pedestrian crossings, number of lanes, traffic count and pedestrian count data (15 minutes count), and the current posted speed limit for each road were collected. The length of Site 1 is 2.78 km, has 8 access points and 3 pedestrian crossings on each direction. The traffic volume is relatively high, but number of pedestrian is low on both weekday and weekend. Site 2 is 0.46 km long with 4 access points and 3 pedestrian crossings on each direction. This site has the highest pedestrian volume with the lowest traffic volume among the four sites. The length of Site 3 is 2.15 km. It has 11 and 8 access points on direction 1 and direction 2, respectively. The site has the highest traffic volume (among the four sites) on both weekday and weekend. Pedestrian volume is also high at this site. Site 4 is 2.94 km long, 4 access points on both directions with no pedestrian crossing. The traffic volume is moderate with very little pedestrian activity. The traffic and pedestrian count data were converted to hourly volume data prior to developing the FLM. It is to be noted that all road attribute data are fixed for each road segments for different time periods and for weekday and weekend except the traffic volume and pedestrian data. The current posted speed limits for site1, site 2, site 3 and site 4 are 40, 60, 80 and 80 km/hr, respectively.

Figure 1 shows the frequency distribution diagrams of observed speeds at site 1, direction 1 for weekday at the time period 6:30-7:30 AM. Speeds higher than the posted speed limit are regarded as "violations". The cumulative percentage of violations (percentage of vehicles exceeding the speed limit) is illustrated in Figure 2.

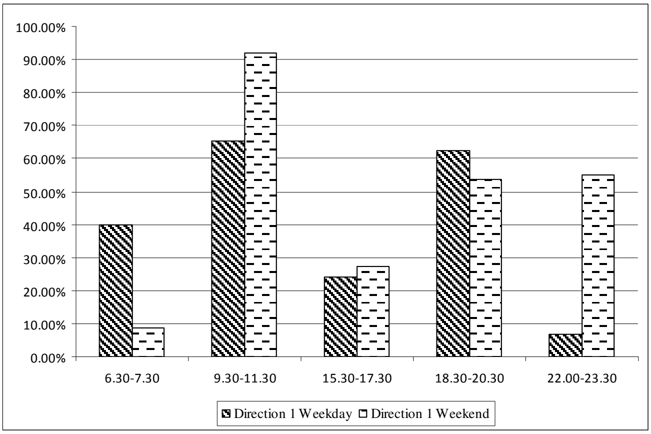


Fig. 2. Posted speed limit violations at site 1 direction 1

3 Development of FLM for 85th Percentile Speed Estimation

The development of the FLM starts with preparing the data sets for both weekday and weekend. The road attribute data collected from the fields were used as the input variables. The estimated 85th percentile of the operating speed was used as the output variable. The input and output variables and their corresponding modified name used in the FLM are shown in Table 1. Two separate models were developed (for weekday and weekend). It is to be noted that volume to capacity ratio was also calculated from the hourly traffic volume to incorporate in the FLM development.

The FLM development was done in two stages using the tool FuzzyTech [21] - first, initial models were developed for both weekday and weekend by setting the memberships (fuzzy sets' parameters) and the knowledge base (rules) intuitively (using some correlation analysis). To overcome the limitations of intuitive setting of knowledge base, the neuro-fuzzy logic (integrated fuzzy and neural nets) was used [17].

Table 1. Input and output variables and their corresponding modified name in fuzzy logic

Variable category	Variable name	Denoted in FLM
Input variables	Length	Length
	Number of access points/intersecting links	IntLnks
	Number of pedestrian crossings	PedCros
	Volume to capacity (V/C) ratio	VCRat
	Hourly pedestrian volume	PedVol
	Posted speed limit	PostSp
Output variable	85th percentile speed	SpEF

3.1 Development of Initial Fuzzy Logic Model

The development of initial models involves three major steps- fuzzification (converting numeric variables into linguistic terms), fuzzy inference (knowledge base- 'IF-THEN' logics) and de-fuzzification (converting linguistic terms into numeric output values).

3.1.1 Fuzzification

The input and output variables are numeric in nature. The drivers mostly perceive these as linguistic terms. For example, the traffic volume may be perceived as high or medium or low rather than its actual numeric values. As such, the numeric values of each input variables were converted into three linguistic terms and the values of the output variable has been converted into five linguistic terms (Table 2). The minimum and maximum values of each variable were determined from the survey results. It is to be noted that the variability of data for the output variable is high and grouping these data into more linguistic terms might result in more accurate estimation of the output variable. On the other hand, three terms have been selected for the input variables due to low variability of the data. It will also reduce the number of rule bases and neuro-fuzzy training time.

Table 2. The proposed FLM variables term definitions

Variable name	Day of the Week	Min	Max	Linguistic terms
Length	Weekday, Weekend	0.46	2.94	Low, medium, high
IntLnks	Weekday, Weekend	4	11	Low, medium, high
PedCros	Weekday, Weekend	0	3	Low, medium, high
VCRat	Weekday	0.08	1.03	Low, medium, high
	Weekend	0.07	1.13	
PedVol	Weekday	0	184	Low, medium, high
	Weekend	0	156	
PostSp	Weekday, Weekend	40	80	Low, medium, high
SpEF	Weekday	13.9	109.9	Very low, low, medium, high, very high
	Weekend	22.36	124.89	

The 'L-shape' membership function (MBF) was used for all variables. The MBFs were generated using the "Compute MBF" fuzzification method. Figure 3 shows the MBF for the Hourly Pedestrian Volume input variable for weekday. For this particular variable, the ranges of linguistic terms were set as (0, 92), (42.465, 138) and (92, 184) for the low, medium and high terms, respectively. The possibility that a numeric level belongs to a linguistic term's range is denoted by the membership degree, μ (Y axis in Figure 3). A μ of 0.0 indicates zero possibility, while μ of 1.0 indicates full membership.

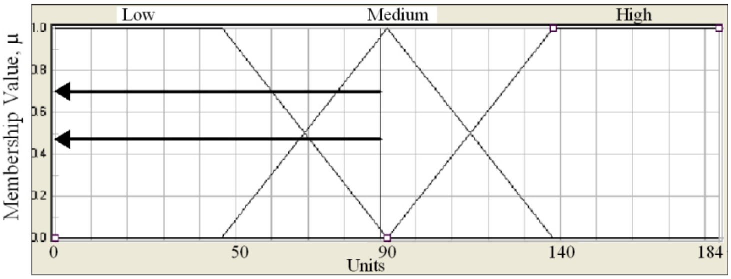


Fig. 3. Membership function for 'hourly pedestrian volume' input variable

3.1.2 Fuzzy Inference (Knowledge Base- 'IF-THEN' Logics)

The rules (IF-THEN logics) were generated to describe the logical relationship between the input variables (IF part) and the output variable (THEN part). The degree of support (DoS) was used to weigh each rule according to its importance. A 'DoS' value of '0' means non-valid rules. Initially, all the DoS's were set to a fixed value of '1'. The IF-THEN rules were formed exhaustively based on the correlation of the input and output variables considering all possible combinations of input and output terms. The neuro-fuzzy training capability was activated in later stage to eliminate non-valid rules (the ones with DoS approaching zero value).

Table 3. Correlation values between input and output variables for both weekday and weekend

	85th percentile speed	
	Weekday	Weekend
Length	0.87	0.82
Number of access points/intersecting links	0.15	0.11
Number of pedestrian crossings	-0.64	-0.35
Volume to capacity (V/C) ratio	0.27	0.08
Hourly pedestrian volume	-0.84	-0.57
Posted speed limit	0.77	0.53

Two correlation matrices were developed for both weekday and weekend to define the relationship between the input and output variables (Table 3) in the fuzzy infer ence system.

It is to be noted that some of the correlation values is showing unexpected signs (e.g. V/C ratio to 85th percentile speed shows positive relation). This is because of Site 2 (a local road), which has very low 85th percentile speed (low posted speed limit of 40 km/hr) and very low traffic volume. Including the data of this particular road segment in calculating the correlation values affects the overall results, particularly because of the limited data (only four segments). Site 2 data were kept for calculating the correlation values to have representation of both road categories in the devised FLM, keeping in mind that increasing the sample road segments may result in better correlation values.

The used operator type for generating the fuzzy rules has been the 'MIN-MAX'. The 'MIN-MAX' method tests the magnitude of each rule and selects the highest one.

The fuzzy composition eventually combines the different rules to one conclusion. The 'BSUM' (Bounded Sum) method was used as it evaluates all rules. A total of 729 rules were generated for both weekday and weekend models. Table 4 shows six rules as an example with the final adjusted DoS's after the neuro-fuzzy training. Detail of the neuro-fuzzy training will be discussed later.

Table 4. Examples of (IF-THEN) rules

IF						THEN	
Length	IntLnks	PedCros	VCRat	PedVol	PostSp	DoS	SpEF
low	low	low	low	low	low	0.90	med.
low	low	low	low	med.	low	1.00	med.
low	low	low	low	high	low	1.00	med.
low	low	low	med.	low	low	1.00	low
low	**low**	**low**	**med.**	**med.**	**low**	**0.90**	**med.**
low	low	low	med.	high	low	0.90	med.

The bold row indicates that for a road segment with low length, low number of intersecting links, low number of pedestrian crossings, medium hourly traffic volume, medium hourly pedestrian volume and low posted speed limit, the estimated 85th percentile speed is medium and the strength for this rule (DoS) is 0.90.

3.1.3 Defuzzification

The results of the inference process are the linguistic terms describing the 85th percentile speed. As mentioned above, five linguistic terms were used for the output results-very low through very high 85th percentile speed). In the defuzzification process, all output linguistic terms are transformed into crisp numeric values. This is done by aggregating (combining) the results of the inference process and then by computing the fuzzy centroid of the combined area. The 'Center-of-Maximum (CoM) method [22] is used for estimating the output numeric value, Y, as follows:

$$Y = \left(\sum_j \mu_{\mathrm{Re}\,sult}(j) * Y_j \right) \Bigg/ \left[\sum_j \mu_{\mathrm{Re}\,sult}(j) \right] \tag{1}$$

Where Y= numeric value representing the 85th percentile speed; μResult(j) = membership value of consequence (linguistic terms) j. Yj is referred to as the base value of the consequence j. It is the consequence's numeric value corresponding to a μ value of 1.

Figure 4 illustrates MBF for the output variable (85^{th} percentile speed) for weekday using the CoM de-fuzzification procedure. The thick arrows indicate the 85^{th} percentile speed base values, Yj on the horizontal axis and the height of the thick black arrows indicate μResult(j). The base values of the five 85^{th} percentile terms are 29.9, 45.9, 61.9, 77.9 and 93.9 respectively. μResult(medium), μResult(high) are 1 and 0.95, respectively. The μResult values of all other terms are zeros. The 85^{th} percentile speed of 69.68 km/hr (indicated by the thin black arrow) was calculated using the Eq. (1).

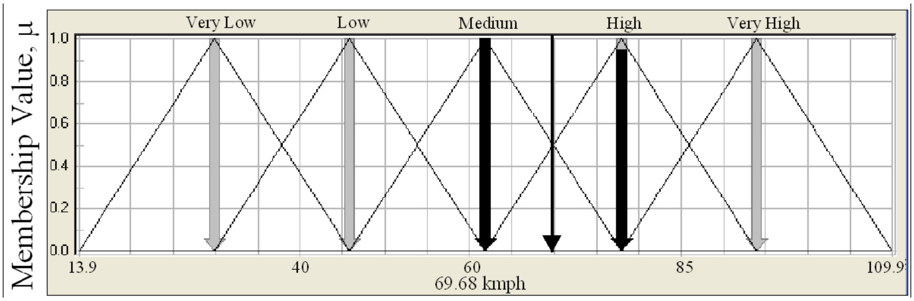


Fig. 4. Membership function for the '85^{th} percentile speed' output variable

3.2 Neuro-fuzzy Data Training

The initial fuzzy logic models for both weekday and weekend were trained in neuro-fuzzy technique. Neuro-fuzzy technique is the combination of neural nets and fuzzy logic. It is comprised of the three fuzzy logic steps (fuzzification, fuzzy inference and de-fuzzification) with a layer of hidden neurons in each process [17]. Fuzzy Associative Maps (FAMs) approach is commonly used in neuro-fuzzy technique to train the data. A FAM is a fuzzy logic rule with an associated weight. This enables the use of a modified error back propagation algorithm with fuzzy logic. The neuro-fuzzy training have been conducted in three steps- defining the MBFs, rules and DoS for training, selection of training parameters, and carrying out training [21].

Initially the default setting of the FuzzyTech tool was used to define range of the numeric values for each term. The rules were formed exhaustively with all DoS values of 1. Therefore in the first step, all MBFs and rules were selected for the neuro-fuzzy training to find the optimized fuzzy logic model. Then the parameters (step width for DoS and terms) were selected for the training. The whole neuro-fuzzy training was carried out for five cycles with each cycle for 1000 iterations.

The step width for the DoS values has been set to 0.1 for each cycle. The step width for the terms has been set to 5% in the first cycle, which was increased by 5% in later cycles. The maximum and average deviations were observed after completion of each cycle. The cycle, for which the deviation values are less, was selected as the final FLM. The process was run for both weekday and weekend models. After the training phase, the MBFs and the DoS values were determined as shown in Table 4 and Figure 5. It can be seen from the Figure 5 that the initial 85th percentile speed terms were set uniformly over the variable's range [Figure 5(a)]. The base value for high 85th percentile speed is 77.9 km/hr (indicated by black arrow). The training algorithm examines the effect of introducing a pre-specified shift to the term's base value (+5% in this case). If the base shift results in a reduction in the deviation, a new base is identified [71.5 in this case as shown with black arrow in Figure 5(b) for weekday model].

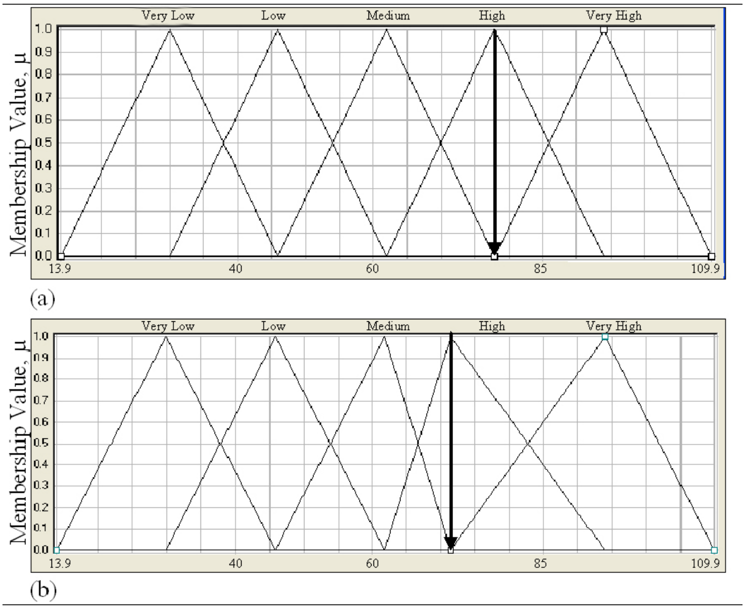


Fig. 5. Membership function of '85th percentile speed' (a) before and (b) after neuro-fuzzy training

4 Model Validation and Result Analysis

After completing the training phase, the 85th percentile speeds were estimated (for both weekday and weekend) with the same set of input data which were used to develop the models. As the notion of fuzzy sets are completely non statistical in nature [23], the residual values (Figure 6) were used to compare both weekday and weekend model results. The x axis of the figure represents a specific road segment and a time period. It can be seen from the figure that the number of positive and negative deviations are almost same for both weekday and weekend models. The maximum deviations for weekday and weekend are 57.63% and 81.44%, respectively. This results in higher average deviation for weekend (19.65% for weekend compared to 14.90% for weekday).

Figure 6 also shows that the number of residuals with values of 15% or less represent 62.5% and 75% of all the residuals for weekday and weekend, respectively. It can be said that both models estimate the 85th percentile speed to a reasonable level for such limited number of sample size.

The estimated values of the 85th percentile speed were classified according to their corresponding current posted speed limits. A comparative descriptive analysis of the estimated (model results) and actual (field data) values of the grouped data for both weekday and weekend models are presented in Table 5.

It is evident in Table 5 that the mean, median, minimum, maximum and standard deviations of the estimated model results are very close to those of the actual data in case of lower posted speed limit (40 km/hr) for both weekday and weekend models. On the other hand, some variations on these values can be observed in both models' results for road segments with higher posted speed limits (60 km/hr and 80 km/hr).

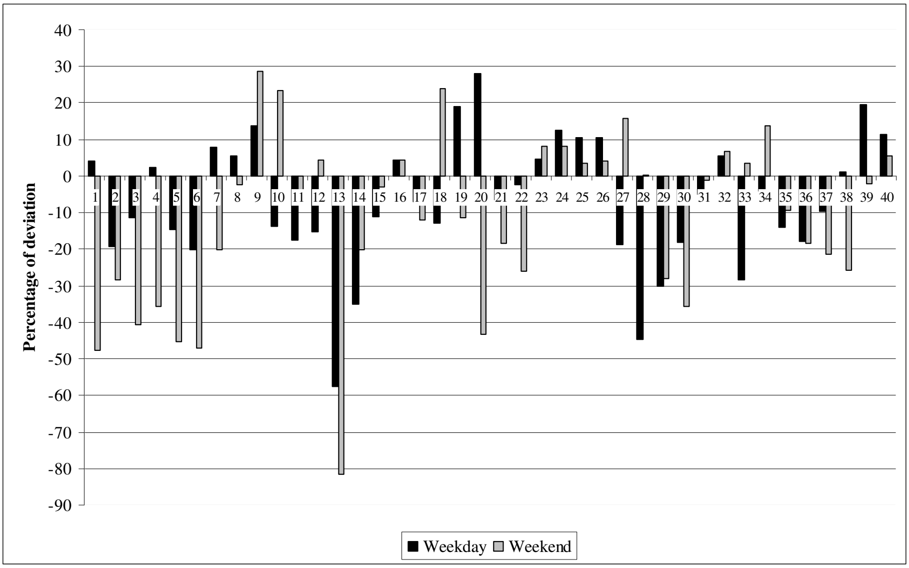


Fig. 6. Percentages of deviations for both weekday and weekend FLMs

Table 5. Comparison of descriptive statistics between the actual field data and the estimated FLM results for both weekday and weekend

				Weekday						Weekend		
	40 km/hr		60 km/hr		80 km/hr		40 km/hr		60 km/hr		80 km/hr	
	S*	E**	S*	E**	S*	E**	S*	E**	S*	E**	S*	E**
Mean	38.2	38.9	67.8	69.7	68.7	75.9	44.8	50.6	84.7	80.9	70.1	83.7
Median	36.6	38.9	66	70	73	74	45.1	51.2	87.1	80.4	71.5	84.2
Min	29.9	29.9	55	68.8	48	71.5	39.5	40.5	52.2	79.7	47	81.6
Max	50	45.9	82	70.7	93.9	81.1	51.5	56.5	107.8	84.2	93	84.2
S.D. ***	6.4	6.1	9.6	0.9	11.4	4.1	4.5	6.3	18.7	1.5	13.6	0.9

*S. Data= Survey Data; **E. Data= Estimated Data; ***S.D.= Standard Deviation.

Figure 7 shows the effects of 'number of pedestrian crossings' and 'length' on the 85th percentile speed for weekday model. As indicated in the figure, the 'length' variable is positively correlated with the 85th percentile speed. On the other hand, the 'number of pedestrian crossings' is negatively correlated with the 85th percentile speed. The highest 85th percentile speed (71.50 km/hr) is found for highest 'length' (2.9 km) and least 'number of pedestrian crossings' (0-1).

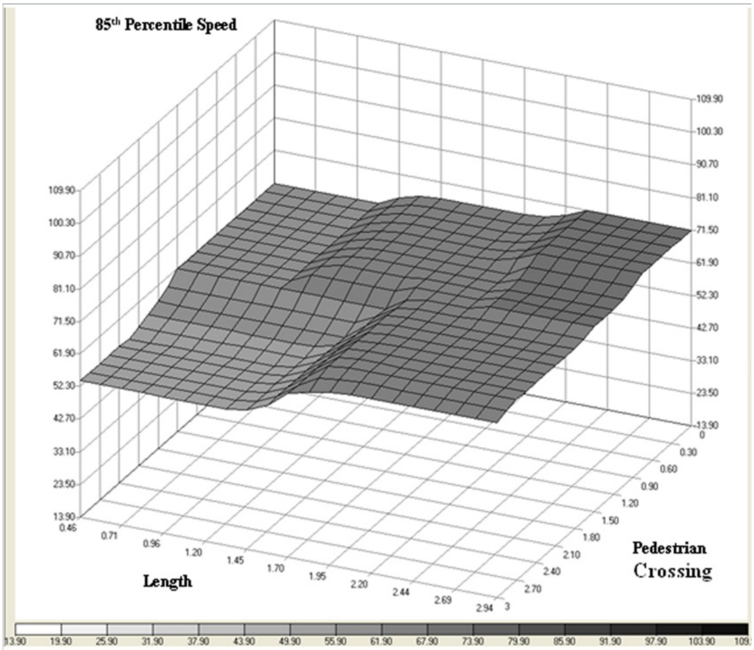


Fig. 7. Effects of 'Length and Number of Pedestrian Crossings' on the '85th Percentile Speed' (weekday model)

Similarly, Figure 8 illustrates the effects of the ‘Posted Speed Limit’ and the ‘Hourly Pedestrian Volume’ (as input variables) on the ‘85th Percentile Speed’ for weekday. As shown, the posted speed limit is positively correlated and hourly pedestrian volume is negatively correlated with the 85th percentile speed. As can also be seen, the effect of the posted speed is not quite noticeable if it exceeds 60 km/hr in cases of high pedestrian volumes.

As shown, the posted speed limit is positively correlated and hourly pedestrian volume is negatively correlated with the 85th percentile speed. As can also be seen, the effect of the posted speed is not quite noticeable if it exceeds 60 km/hr in cases of high pedestrian volumes.

It can be said that regardless limited number of data, fuzzy logic shows the relationship between the input and output variables realistically. As fuzzy logic handles linguistic terms (for a range of numeric values), it is less sensitive to each individual numeric value. This replicates true human nature about perceiving factors on the roads. For example, it is clear that drivers’ choice of operating speed (represented by 85th percentile speed) is influenced by the length of the road segment or pedestrian volume. With larger length, the operating speed tends to be higher. Such changes do not occur for every one km change of length. In reality, the decision of choosing any particular range of operating speed tend to be stable for range of length (say between 0 to 1 km). Fuzzy logic predicts such relationship very realistically.

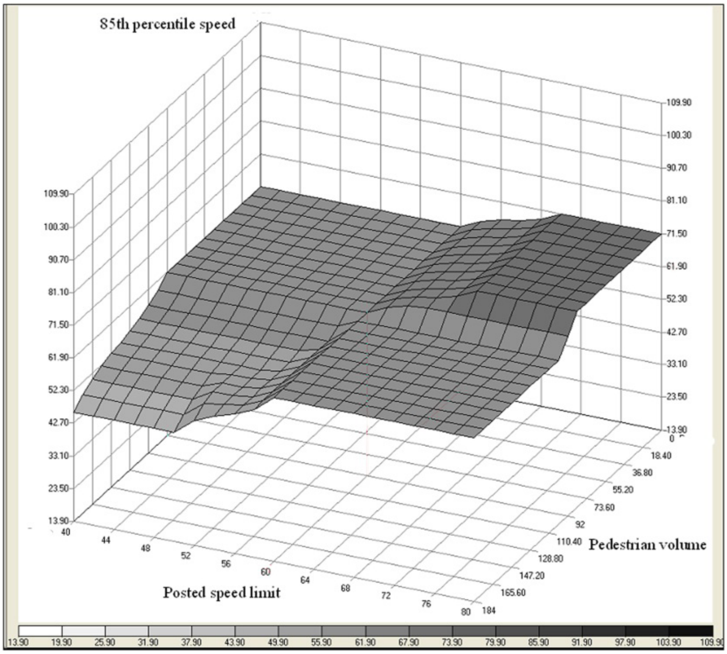


Fig. 8. Effects of ‘Posted Speed Limit and Hourly Pedestrian Volume’ on the ‘85th Percentile Speed’ (weekday model)

5 Conclusions

This paper discussed the development of the FLM for estimating the 85th percentile speed based on six road's attribute data. The advantage of fuzzy logic is its ability to address the uncertain nature of human thinking (perception). The same road (road attribute data) can be perceived differently by different drivers and choose their operating speed accordingly. The other advantage is the using the neuro-fuzzy which can be utilized to automate the development of the knowledge base.

The FLMs are widely known for describing the vagueness and nonlinearity in the human behavior relationships between inputs and output. However, such models are generally only valid in situations for which data are available to calibrate the model. If the FLM is to be used to assess the choice behavior that is not covered in the data for calibration, the applicability of the model for estimating the 85th percentile speed might be questionable. As such, the data for calibration should thoroughly cover the entire range of (input and output) variables for better and more accurate estimation.

Identifying and setting appropriate posted speed limit for a given road segment is a complex task which involves studying the speed behavior pattern of the drivers, the characteristics of road environment, road geometry, etc. This study focused on only one aspect; the drivers' speeding behavior based on the basic road characteristics, the traffic intensity and pedestrian activities for a very limited number of road segments.

One may argue the necessity to develop such models while such 85th percentile speed can be actually measured in the field. In response to such argument is that tremendous savings in the resources (that would be needed to carry on actual field survey measures over an entire network) can be materialized. It is envisioned that these models can be developed with a reasonable representative sample of road segments in a typical network. The derived models can then be validated and subsequently applied to the entire network.

Keeping in mind the limited data set used in the study (due to the resources constraints), that likely contributes to deficiencies in representing the various road characteristics and environmental factors (with only few data points); it is legitimate to assume that the richness in data collection will ultimately lead to better more statistically significant models. Along this line, it is suggested that a systematic sampling approach should be adopted in selecting the road segments to include in the data set to use for models' calibration. The principles of the minimum sample size should be observed. It is suggested that a stratified sampling procedure to be used in selecting the road segments for spot speed field observations. All the network roadway segments may be stratified based on their intrinsic characteristics of posted speed, length, traffic volume, pedestrian intensity, etc. A representative stratified sampling procedure with a minimum sample size according to a pre-specified confidence level and interval should be observed in generalizing the fuzzy logic modeling approach.

Acknowledgements. This research is part of M.Sc. thesis entitled 'Assessing the Methodology of Setting Posted Speed Limit in Al Ain-UAE' funded by Roadway, Transportation and Traffic Safety Research Center, United Arab Emirates University.

References

1. American Association of State Highway and Transportation Officials (AASHTO), Synthesis of Speed Zoning Practice. Technical Summary, Report No. FHWA/RD-85/096, Subcommittee on Traffic Engineering, Federal Highway Administration, US Department of Transportation (1985)
2. Department for Transportation (DfT), Setting Local Speed Limits (2006), `http://www.dft.gov.uk/pgr/roadsafety/speedmanagement/dftcircular106/dftcircular106.pdf` (retrieved)
3. GRSP (Global Road Safety Partnership), Speed Management: A Road Safety Manual for Decision Makers and Practitioners. World Health Organization, World Bank (2008)
4. Transportation Research Board (TRB), Managing Speed: Review of Current Practice for Setting and Enforcing Speed Limit. Special Report No. 254, National Research Council, Washington, D.C. (1998)
5. Srinivasan, R., Parker, M., Harkey, D., Tharpe, D., Summer, R.: Expert System for Recommending Speed Limits in Speed Zones. Final Report, Project No. 3-67, NCHRP, National Research Council (2006)
6. Fitzpatrick, K., Carlson, P., Wooldridge, M.D., Miaou, S.: Design Speed, Operating Speed, and Posted Speed Practices. NCHRP Report 504, Transportation Research Board, National Research Council, Washington, D.C. (2003)
7. Minnesota Department of Transportation (MNDOT), Speed Limit? Here is your answer. Office of Traffic Engineering and Intelligent Transportation System, Minnesota Department of Transportation (2002), `http://www.dot.state.mn.us/speed/~SpeedFlyer2002.pdf` (retrieved)
8. Poe, C.M., Mason, J.M.: Analyzing Influence of Geometric Design on Operating Speeds along Low-Speed Urban Streets: Mixed-Model Approach. Transportation Research Record 1737, 18–25 (2000)
9. Haglund, M., Aberg, L.: Speed Choice in Relation to Speed Limit and Influences from Other Drivers. Transportation Research Part F: Traffic Psychology and Behavior 3, 39–51 (2000)
10. Fitzpatrick, K., Carlson, P., Brewer, M., Wooldridge, M.: Design Factors That Affect Driver Speed on Suburban Streets. Transportation Research Record 1751, 18–25 (2001)
11. Wang, J.: Operating Speed Models for Low Speed Urban Environments Based on In-Vehicle Data. Ph. D. Georgia Institute of Technology (2006)
12. Fildes, B.N., Fletcher, M.R., Corrigan, J.McM.: Speed Perception 1: Driver's Judgments of Safety and Travel Speed on Urban and Rural Straight Roads. Report CR54, Federal office of road safety, Department of transport and communications, Canberra (1987)
13. Fildes, B.N., Leening, A.C., Corrigan, J.McM.: Speed Perception 2: Driver's Judgments of Safety and Travel Speed on Rural Curved Roads and at Night. Report CR60, Federal office of road safety, Department of transport and communications, Canberra (1989)
14. Tignor, S.C., Warren, D.: Driver Speed Behavior on U.S. Streets and Highways. Compendium of Technical Papers, Institute of Transportation Engineers, Washington, D.C. (1990)
15. Tarris, J.P., Poe, C.M., Mason, J.M., Goulias, K.G.: Predicting Operating Speeds on Low-Speed Urban Streets: Regression and Panel Analysis Approaches. Transportation Research Record 1523, 46–54 (1996)
16. Zadeh, L.A.: The role of fuzzy logic in the management of uncertainty in expert systems. Fuzzy Sets and Systems 11(1-3), 197–198 (1983)
17. Hawas, Y.E.: Development and Calibration of Route Choice Utility Models: Neuro-Fuzzy Approach. Journal of Transportation Engineering 130(2) (2004)

18. Kikuchi, S., Pursula, M.: Treatment of Uncertainty in Study of Transportation: Fuzzy Set Theory and Evidence Theory. Journal of Transportation Engineering 124(1), 1–8 (1998)
19. Zimmermann, H.J.: Fuzzy Set Theory and Its Applications. Kluwer Academic, USA (1996)
20. Ewing, R.: Traffic Calming Impacts. In: ITE (Institute of Transportation Engineers) Traffic Calming: State and Practice, Washington, D.C., pp. 99–126 (1999)
21. INFORM, FuzzyTech 5.5: User's Manual. GmbH/Inform Software Corporation, Germany (2001)
22. Ross, T.J.: Fuzzy logic with engineering applications. McGraw-Hill, New York (1995)
23. Zadeh, L.A.: Fuzzy Sets. Information and Control 8, 338–353 (1965)

Language Independent Extraction of Key Terms: An Extensive Comparison of Metrics

Luís F.S. Teixeira[2], Gabriel P. Lopes[2], and Rita A. Ribeiro[1]

[1] CA3-Uninova, Campus FCT/UNL, 2829-516 Caparica, Portugal
[2] CITI, Dep. Informática, FCT/UNL, 2829-516 Caparica, Portugal
lst@luisteixeira.org, gpl@fct.unl.pt, rar@uninova.pt

Abstract. In this paper twenty language independent statistically-based metrics used for key term extraction from any document collection are compared. Some of those metrics are widely used for this purpose. The others were recently created. Two different document representations are considered in our experiments. One is based on words and multi-words and the other is based on word prefixes of fixed length (5 characters for the experiments made). Prefixes were used for studying how morphologically rich languages, namely Portuguese and Czech behave when applying this other kind of representation. English is also studied taking it, as a non-morphologically rich language. Results are manually evaluated and agreement between evaluators is assessed using k-Statistics. The metrics based on Tf-Idf and Phi-square proved to have higher precision and recall. The use of prefix-based representation of documents enabled a significant precision improvement for documents written in Portuguese. For Czech, recall also improved.

Keywords: Document keywords, Document topics, Words, Multi-words, Prefixes, Automatic extraction, Suffix arrays.

1 Introduction

A key term, a keyword or a topic of a document is any word or multi-word (taken as a sequence of two or more words, expressing clear cut concepts) that reveals important information about the content of a document from a larger collection of documents.

Extraction of document key terms is far from being solved. However this is an important problem that deserves further attention, since most documents are still (and will continue to be) produced without explicit indication of their key terms as metadata. Moreover, most existing key term extractors are language dependent and, as such, require linguistic processing tools that are not available for the majority of the human languages. Hence our bet, namely in this paper, is on language independent extraction of key terms.

So, the main aim of this work is to compare metrics for improving automatic extraction of key single and multi-words from document collections, and to contribute to the discussion on this subject matter.

Our starting point was the work by [1], on multiword key term extraction, where two basic metrics were used: Tf-Idf and relative variance (Rvar). By looking more

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 69–82, 2013.

carefully at the examples shown in [1], where plain Tf-Idf metric is used, it became apparent that, the comparison made between the two metrics was unfair. A fair comparison would require the use of a Tf-Idf derived metric taking into account the Tf-Idf values for multi-word extremities as well as the medium character length per word of each multi-word as it had been proposed for the use of Rvar variant metric in [1]. Moreover, as one needs to calculate the relevance of words and multi-words using the same metrics, we decided to rank simultaneously words and multi-words describing the content of any document in a collection according to the metric assigned value to that word or multi-word. This diverges from the proposal made in [2] where an "a priori" fixed proportion of words and multi-words is required. And no one knows "a priori" which documents are better described by words alone or by multi-words. Nor does she/he know the best proportion of key words or key multi-words.

This way, our work improves the discussion started in [1], and continued in [2], but we arrive at different conclusions, namely that Tf-Idf and Phi-square based metrics enabled higher precision and recall for the extraction of document key terms. The use of a prefix-based representation of documents enabled a significant improvement for documents written in Portuguese and a minor improvement for Czech, as representatives of morphologically rich languages, regarding precision results.

Additionally we also extend the preliminary discussion started in [3] where some of the metrics used in current work were presented. To achieve our aims we compare results obtained by using four basic metrics (Tf-Idf, Phi-square, Mutual Information and Relative Variance) and derived metrics taking into account per word character median length of words and multi-words and giving specific attention to word extremities of multi-words and of words (*where left and right extremities of a word are considered to be identical to the word proper*). This led to a first experiment where we compare 12 metrics (3 variants of 4 metrics). On a second experiment, we decided to use a different document representation in terms of word prefixes of 5 characters in order to tackle morphologically rich languages. As it would be senseless to evaluate the relevance of prefixes, it became necessary to project (bubble) prefix relevance into words and into multi-words.

All the experimental results were manually evaluated and agreement between evaluators was assessed using k-Statistics.

This paper is structured as follows: related work is summarized in section 2; our system, the data and the experimental procedures used are described in section 3; the metrics used are defined in section 4; results obtained are shown in section 5; conclusions and future work are discussed in section 6.

2 Related Work

In the area of document classification it is necessary to select features that later will be used for training new classifiers and for classifying new documents. This feature selection task is somehow related to the extraction of key terms addressed in this paper. In [4], a rather complete overview of the main metrics used for feature selection for document classification and clustering is made.

As for the extraction of multi-words and collocations, since we also need to extract them, we just mention the work by [5], using no linguistic knowledge, and the work by [6], requiring linguistic knowledge.

In the area of keyword and key multi-word extraction, [7-11] address the extraction of keywords in English. Moreover those authors use language dependent tools (stop-words removing, lemmatization, part-of-speech tagging and syntactic pattern recognition) for extracting noun phrases. Being language independent, our approach clearly diverges from these ones. Approaches dealing with extraction of key-phrases (that are according to the authors "*short phrases that indicate the main topic of a document*") include the work of [12] where Tf-Idf metric is used as well as several language dependent tools. In [13], a graph-based ranking model for text processing is used. The authors use a 2-phase approach for the extraction task: first they select key-phrases representative of a given text; then they extract the most "important" sentences in a text to be used for summarizing document content.

In [14] the author tackles the problem of automatically extracting key-phrases from text as a supervised learning task. And he deals with a document as a set of phrases, which his classifier learns to identify as positive or negative examples of key-phrases.

[15] deal with eight different languages, use statistical metrics aided by linguistic processing, both to extract key phrases and keywords. Dealing also with more than one language, [1] extract key multi-words using purely statistical measures. In [2] statistical extraction of keywords is also tackled but a predefined ratio of keywords and key multi-words is considered per document, thus jeopardizing statistical purity.

[16] present, a keyword extraction algorithm that applies to isolated documents, not in a collection. They extract frequent terms and a set of co-occurrences between each term and the frequent terms.

In summary, the approach followed in our work is unsupervised, language independent and extracts key words or multi-words solely depending on their ranking values obtained by applying the 20 metrics announced and explained bellow in section 4.

3 System Data and Experiments

Our system is made of 3 distinct modules. First module is responsible for extracting multi-words, based on [17] and using the extractor of [18]. A Suffix Array was used [19,20] for frequency counting of words, multi-words and prefixes. This module is also responsible for ranking, according to the metrics used, words and multi-words per document. And it allows the back office user to define the minimum word and prefix length to be considered. In the experiments reported we have fixed minimum word length at 6 characters and minimum prefix length a5 characters.

Second module is a user interface designed to allow external evaluators to classify the best 25 terms ranked according to each of selected metrics. When moving from ranking classification related to one metric to the ranking produced by another metric, evaluations already made are pervasively propagated. This feature enables evaluators to reconsider at any time some of their own earlier options.

Third module is also a user interface acting as a back office application, allowing an easy interpretation of the classifications produced by the external evaluators. It also shows graphically the k-Statistics resulting from evaluations of any two evaluators.

We worked with a collection of texts, for the three languages experimented, Portuguese (pt), English (en) and Czech (cs), from European legislation (*http://eur-lex.europa.eu/[L]/legis/latest/chap16.htm*, where *[L]* can be replaced by any of the following language references: pt, en or cs). The texts were about Science, Dissemination of information and Education and Training. Czech corpus also included texts about Culture. Apart from these texts for Czech, the remaining of the corpus documents was parallel for the three languages. So the total number of terms for these collections was: 109449 for Portuguese, 100890 for English and 120787 for Czech.

We worked with words having a minimum length of six characters (this parameter is changeable) and filtered multi-words (with words of any length) by removing those containing punctuation marks, numbers and other special symbols. As it will be seen later some additional filtering operations will be required and discussed in the conclusions section.

Evaluators were asked to evaluate the 25 best ranked terms for a selected sub-group of six of the twenty metrics for a sub-set of five randomly selected documents of a total of 28 documents for each language. The Evaluators had access to the original documents from where key-words were extracted. When document metadata contained the keywords assigned to it, evaluators had also access to this information thus helping the evaluation task. It is worth telling that when this metadata exists, it generally does not use mutatis mutantis the multi-word terms as they are used in the document. This information was not used for the extraction task performed.

Four classifications were considered in the evaluation: “good topic descriptor” (G), “near good topic descriptor” (NG), bad topic descriptor” (B), and “unknown” (U). A fifth classification, “not evaluated” (NE), was required to enable the propagation of evaluation, for those metrics that were not specifically evaluated. In Section 5 the results of the experiments are presented.

It must be stressed that the multiword extractor used is available on the web page referred in [18].

4 Metrics Used

As mentioned before, we used 4 basic metrics: Tf-Idf, Phi-square, Relative Variance (Rvar) and Mutual Information (MI).

Formally, **Tf-Idf** for a term t in a document *dj* is defined in equations (1), (2) and (3).

$$Tf - Idf\ (t, d_j) = p(t, d_j) * Idf\ (t, d_j) \tag{1}$$

$$p(t, d_j) = f(t, d_j)/Nd_j \tag{2}$$

$$Idf(t, d_j) = \log(\|D\|/\|\{d_j : t \in d_j\}\|) \tag{3}$$

Notice that, in (1), instead of using the usual term frequency factor, probability $p(t, d_j)$, defined in equation (2), is used. There, $f(t, d_j)$, denotes the frequency of term t in document *dj*, t denotes a prefix, a word, or a multiword, and *Ndj* refers to the number of words or n-grams of words contained in d_j. The total number of documents present in the corpus is given by $\|D\|$. The use of a probability in (1) normalizes the Tf-Idf metric, making it independent of the size of the document under consideration.

Rvar (Relative Variance) is the metric proposed by [1], defined in equation (4). It does not take into account the occurrence of a given term *t* in a specific document in the corpus. It deals with the whole corpus, and thus loses the locality characteristics of term t. This locality is recovered when the best ranked terms are reassigned to the documents where they occur.

$$\mathrm{Rvar(t)} = (1/\|\mathrm{D}\|) * \sum_{\mathrm{j}} \left(\mathrm{p}(\mathrm{t}, \mathrm{d_j}) - \mathrm{p(w,.)/p(t,.)}\right)^2 \tag{4}$$

$p(t, d_j)$ is defined in (2) and $p(t,.)$denotes the mean probability of term *t*, taking into account all documents in the collection. As above, we take *t* as denoting a prefix, a word, or a multiword.

Phi-Square [21] is a variant of the well-known Chi-Square metric. It allows a normalization of the results obtained with Chi-Square, and is defined in equation (5), where M is the total number of terms (prefixes, words, or multi-words) present in the corpus (the sum of terms from all documents in the collection). A denotes the number of times term t occurs in document d. B stands for the number of times term $t$occurs in documents other than d, in the collection. C means the number of terms of the document d subtracted by the amount of times term t occurs in document$d$. Finally, D is the number of times that neither document d nor term t co-occur (i.e., N-A-B-C, where N denotes the total number of documents).

$$\varphi^2(t,d) = \frac{\left(\frac{N * (AD - CB)^2}{(A + C) * (B + D) * (A + B) * (C + D)}\right)}{M} \tag{5}$$

Mutual Information [22] is a widely used metric for identifying associations between randomly selected terms. For our purposes we used equation (6) where t, d, A, B, C and N have identical meanings as above for equation (5).

$$MI(t,d) \approx \log(A * N/(A + C) * (A + B)) \tag{6}$$

In the rest of this section we will introduce derivations of the metrics presented above for dealing, on equivalent grounds, with aspects that were considered crucial in [1,2] for extracting key terms. Those derivations will be defined on the basis of 3 operators: Least (L), Median (M) and Bubble (B). In the equations below MT stands for any of the previously presented metrics (Tf-Idf, Rvar, Phi-square or φ^2, and Mutual Information or MI), P stands for a Prefix, W for a word, $MW$for a multi-word taken as word sequence $(w_1 \ldots w_n)$.

Least Operator is inspired by the metric LeastRvar introduced in [1] and coincides with that metric if it is applied to Rvar.

$$LeastRvar(MW) = \min\left(Rvar(w_1), Rvar(w_n)\right) \tag{7}$$

$LeastRvar(MW)$ is determined as the minimum of $Rvar$ applied to the leftmost and rightmost words of MW, $\mathrm{w_1}$ and $\mathrm{w_n}$. In order to treat all metrics on equal grounds operator “Least” will now be applied to metric *MT*, where *MT* may be any of the metrics Tf-Idf, Rvar, φ^2, and MI as depicted in equation (9), when a multiword MW is at stake. As above, Least_MT of a multiword MW will be equal to the minimum of the MT metric value for the extremity words, w_1 or w_n, in the multi-word MW. This operator was adapted to work with words alone as in equation (8), where the Least_MT for a word W is identical to the rank value obtained for that word using metric MT. Adaptation was made by assuming that the leftmost and rightmost words of a single word coincide with the word itself.

$$Least\_MT\ (W) = MT(W) \tag{8}$$

$$Least\_MT\ (MW) = min\big(MT(w_1), MT(w_n)\big) \tag{9}$$

Bubbled Operator, another problem we needed to solve was the propagation of the relevance of each Prefix (P) to words (W) having P as a prefix.

$$Bubbled\_MT\ (W) = MT(P) \tag{10}$$

$$Least\_Bubbled\_MT\ (MW)\ =\ min\big(Bubbled\_MT(w_1), Bubbled\_\ MT(w_n)\big) \tag{11}$$

In bubble based metrics, the rank of a prefix is assigned to the words it prefixes. Generally it is larger than the rank assigned by the corresponding metric to the word forms it prefixes. For example, the value assigned to the 5 character prefix “techn” in a text would be propagated to all words having that prefix, namely “technology”, “technologies”, “techniques”, if they would appear in the same text.

Median Operator was added in order to better compare the effects of using an operator similar to the one proposed in [2] which took into account the median character length of words in multi-words. By doing so, we got metrics defined in equations (12) and (13), where *T* represents a term (word or multi-word), *LM* stands for *Least_Median* operator applied to any base metric MT and *LBM* stands for *Least_Bubble_Median operator applied to metric MT.* And *Median* of a term *T* is the median of character lengths of words in a multi-word or of the word at stake. For example, for a multiword made of three words, of lengths 5, 2 and 6, median length is 5.

$$LM\_MT(T) = Least\_MT(T) * Median(T) \tag{12}$$

$$LBM\_MT\ (T) = Least\_Bubble\_MT(T) * Median(T) \tag{13}$$

5 Results

In this section we present some of the results obtained. We will also show that Rvar and its related metrics behave worse than the ones based on Tf-Idf and Phi Square, contradicting results presented in the work of [1].

An example of the top five terms extracted from one document, ranked by the Phi-Square metric for the worked languages is shown in Table 1. This document was about scientific and technical information and documentation and ethics.

As the corpus used elaborated on Science, Information dissemination, education and training, for the example document the word “science” alone was naturally demoted.

It is important to notice also that documents were in many cases short. This has a direct impact on results, as the number of relevant words and multi-words is short and most of them are irrelevant in terms of document content. As a consequence precision obtained for shorter documents is lower than for longer documents as most of the times just one term describes document content. Longer documents pose not this problem.

Table 1. Top terms ranked by Phi-Square metric, manually classified as Good (G), Near Good (NG) or Bad (B), for 3 languages for a document on scientific and technical information and documentation.

Portuguese	English	Czech
ciências e as novas tecnologias (G)	group on ethics (G)	skupiny pro etiku ve vědě (G)
ciências e as novas (B)	ethics (G)	nových technologiích (NG)
ética para as ciências (G)	science and new technologies (G)	etiku ve vědě (G)
grupo europeu de ética (G)	the ege (G)	skupiny pro etiku (G)
membros do gee (G)	ethics in science (G)	vědě a nových technologiích (NG)

In the previous table, some top-ranked key terms are a sub terms of others. This has some effect on the results, because they are not mutually independent. Looking more carefully we may also notice larger, more specific, multi-words that might be rather sharp descriptors of document content as would be the case of "group on ethics in science and new technologies". We will return to this discussion on section 6.

For the same document, best performing metric based on Rvar (see Table 3) LBM_Rvar just extracted "ethics" in position 20. Other extracted top terms include names of several European personalities, having as such very poor extraction results.

In tables 2 and 3, average precision values obtained for the 5, 10 and 20 best ranked key terms extracted using different metrics are shown.

Regarding recall values presented in tables 4 and 5, it is necessary to say that: 1) Tf-Idf, Phi Square and derived metrics extract very similar key terms; 2) Rvar and MI, alone, are unable to extract key terms as, depending on the length of documents, the top ranked 100, 200 or more terms are equally valuated by these metrics; 3) derived metrics of Rvar and MI extract very similar rare key terms completely dissimilar from those extracted by Tf-Idf, Phi Square and derived metrics; 4) by evaluating the 25 best ranked terms by 6 metrics (Phi Square, Least Tf-Idf, Least Median Rvar, Least Median MI, Least Bubble Median Phi Square and Least Bubble Median Rvar) we obtained from 60 to 70 terms evaluated per document.

Recall was determined on the basis of these 60 to 70 evaluated terms. So, recall values presented in tables (4) and (5) are upper bounds of real recall values. Table 2 shows results for the metrics with best precision for the three languages, all of them with results above 0.50. Notice, that for Portuguese and Czech, the average precision is similar. The best results were obtained for the top ranked 5 terms, decreasing with similar values when dealing with the top ranked 10 and 20 terms. In average, English language presents the best results.

Also from Table 2 we can point out that, for Portuguese, best results were obtained with metrics *Least Bubble Tf-Idf* and *Least Bubble Median Tf-Idf.* This means that Bubble operator and prefix representation enabled precision results closer to those obtained for English. Tf-Idf had the best results in Czech, for all thresholds. In English, *Least Median Phi Square* enabled the best results. Moreover, for the 10 best terms threshold, English has three metrics with the best results, the one already mentioned and *Least Median Tf-Idf and Phi-Square*.

As pointed above, Rvar and MI metrics alone were unable to discriminate the top 5, 10 or 20 best ranked terms. This probably explains the need to use the Least and Median operators proposed by [1,2]. Precision for the Rvar and MI derived metrics is shown in table 3. It shows clearly that Tf-Idf and Phi Square based metrics, in table 2, are much better than those based on Rvar and MI. They get for the best metrics, values a bit higher than 0.50, and generally all bellow 0.50 which makes the average precision for these metrics rather poor.

Table 2. Average precision values for the 5, 10 and 20 best terms using the best metrics, and average for each threshold

Czech			
Metric	Prec. (5)	Prec. (10)	Prec. (20)
Tf-Idf	0.90	0.86	0.66
L Tf-Idf	0.75	0.70	0.61
LM Tf-Idf	0.70	0.65	0.59
LB Tf-Idf	0.80	0.68	0.65
LBM Tf-Idf	0.65	0.68	0.66
φ2	0.70	0.70	0.61
L φ2	0.70	0.60	0.58
LM φ2	0.70	0.60	0.58
LB φ2	0.55	0.63	0.55
LBM φ2	0.55	0.65	0.59
Average	**0.72**	**0.68**	**0.61**
English			
Metric	Prec. (5)	Prec. (10)	Prec. (20)
Tf-Idf	0.84	0.74	0.67
L Tf-Idf	0.78	0.66	0.68
LM Tf-Idf	0.81	0.78	0.66
LB Tf-Idf	0.85	0.66	0.65
LBM Tf-Idf	0.82	0.69	0.62
φ2	0.84	0.78	0.68
L φ2	0.83	0.76	0.69
LM φ2	0.87	0.78	0.70
LB φ2	0.83	0.74	0.62
LBM φ2	0.80	0.74	0.65
Average	**0.83**	**0.73**	**0.66**
Portuguese			
Metric	Prec. (5)	Prec. (10)	Prec. (20)
Tf-Idf	0.69	0.70	0.66
L Tf-Idf	0.64	0.66	0.65
LM Tf-Idf	0.68	0.63	0.64
LB Tf-Idf	0.86	0.71	0.65
LBM Tf-Idf	0.83	0.70	0.68
φ2	0.73	0.73	0.62
L φ2	0.68	0.64	0.59
LM φ2	0.61	0.64	0.59
LB φ2	0.60	0.65	0.65
LBM φ2	0.62	0.61	0.62
Average	**0.70**	**0.67**	**0.63**

In terms of "Recall" (upper bounds of recall), shown in tables 4 and 5, one of our goals was to increase the Czech recall, which we believe to have accomplished. In the same line with precision, the metrics based on Tf-Idf and Phi-Square have better recall values, in table 4, than those obtained for Rvar and MI-based metrics, in table 5. We have chosen to present "recall" values for the top 20 ranked relevant terms as these values are higher than for 5 or 10 best ranked terms. Recall values obtained for Rvar and MI derived metrics (Table 5) are much lower than those obtained for Tf-Idf and Phi-Square derived metrics, as Rvar and MI derived metrics choose rare terms that may specify very specific subject matters of documents.

Table 3. Precision values for the 5, 10 and 20 best terms using the Rvar and MI best metrics, and average for each threshold

Czech			
Metrics	Prec. (5)	Prec. (10)	Prec. (20)
LBM Rvar	0.50	0.39	0.27
LM Rvar	0.45	0.31	0.22
LBM MI	0.40	0.40	0.26
LM MI	0.45	0.31	0.22
Average	**0.45**	**0.35**	**0.24**
English			
Metrics	Prec. (5)	Prec. (10)	Prec. (20)
LBM Rvar	0.52	0.43	0.40
LM Rvar	0.47	0.42	0.35
LBM MI	0.46	0.49	0.43
LM MI	0.47	0.42	0.34
Average	**0.48**	**0.44**	**0.38**
Portuguese			
Metrics	Prec. (5)	Prec. (10)	Prec. (20)
LBM Rvar	0.52	0.48	0.41
LM Rvar	0.46	0.36	0.35
LBM MI	0.52	0.48	0.43
LM MI	0.42	0.35	0.33
Average	**0.48**	**0.42**	**0.38**

Table 4. "Recall" Values for threshold of 20 best terms for Tf-Idf and Phi Square based metrics, and average recall

	Czech	**English**	**Portuguese**
	P(20)	P(20)	P(20)
tfidf	0.68	0.43	0.48
L Tf-Idf	0.56	0.48	0.46
LM tfidf	0.52	0.43	0.44
LB tfidf	0.60	0.38	0.37
LBM tfidf	0.54	0.35	0.40
φ2	0.50	0.44	0.48
L φ2	0.50	0.41	0.36
LM φ2	0.51	0.43	0.37
LB φ2	0.40	0.37	0.33
LBM φ2	0.43	0.41	0.35
Average	**0.54**	**0.41**	**0.40**

Table 5. Recall Values for threshold of 20 best terms for Rvar and MI based metrics, and average recall

	Czech	English	Portuguese
	P(20)	P(20)	P(20)
LBM Rvar	0.20	0.16	0.13
LM Rvar	0.25	0.12	0.14
LBM MI	0.20	0.16	0.15
LM MI	0.25	0.11	0.13
Average	**0.23**	**0.14**	**0.14**

Table 6. Kappa statistics-based agreement between the evaluators, for Portuguese and English, for Tf-Idf and Phi-Square based metrics

	Portuguese	English
tfIdf	0.57	0.35
LM tfidf	0.56	0.42
LB tfidf	0.67	0.38
LBM tfidf	0.64	0.40
L φ2	0.64	0.46
LM φ2	0.56	0.40
LBM φ2	0.54	0.31

Tables 6 and 7 depict the agreement between evaluators, for Portuguese and English, by using Kappa statistics. It shows that for Portuguese we have higher levels of agreement for the Tf-Idf and Phi-Square based metrics. For English agreement achieved is not so high, but never the less, we consider it acceptable.

Disagreement was mainly due to acceptance of some adjectives as near good descriptors by one of the evaluators, while the other systematically rejected them in the sense that adjectives, by themselves, are not good descriptors. This means that, if the evaluation phase had been preceded by identification of a couple of cases where the evaluation would be dissimilar, the agreement obtained would have been higher. Disagreement regarding Rvar and MI based metrics occurred mainly because selected key terms occurred just once and it was very hard to agree on how such rare terms could be key terms of those documents. We have not achieved to have the results for Czech evaluated by two persons. But it should be mentioned that Czech poses yet another problem when evaluation is at stake, due to its richer morphology. For the example shown in table 1, one observes that multi-words extracted and ranked are mostly sub-phrases of multi-word "group on ethics in science and new technologies" if not of the 11-word term "members of the group on ethics in science and new technologies". While for Portuguese and English this has almost no consequences, for Czech, "skupiny pro etiku ve vědě" is a translation of "of group on ethics in science" which is not exactly a term. Corresponding term in nominative case would be "skupina pro etiku ve vědě". It was accepted as adequate (G) as it also translates as

Table 7. Kappa statistics-based agreement between the evaluators, for Portuguese and English, for Rvar and MI based metrics

	Portuguese	**English**
LBM rvar	0.28	0.24
LM rvar	0.27	0.28
LBM MI	0.07	0.28
LM MI	0.19	0.22

"groups on ethics in science" that is not present in the Portuguese and English versions of the same text. Similarly, "etiku ve vědě" is the accusative case for "etika ve vědě". Results obtained enable however a clear idea about the content of the document. But evaluation, for languages as Czech and other languages having word forms modified by case, still need to be deeply discussed or may require a post extraction normalizer to bring phrases to nominative case.

6 Conclusions and Future Work

Our approach to key-term extraction problem (of both words and multi-words) is language independent.

By ranking separately words and multi-words, using 20 metrics, based on 4 base metrics, namely Tf-Idf, Phi Square, Rvar (relative variance) and MI (Mutual Information), and by merging top ranked words' list with top ranked multi-words' list taking into account the values assigned to each word and multi-word by each of the metrics experimented we were able to make no discrimination between words and multi-words, as both entities pass the same kind of sieve/metrics to be ranked as adequate key-terms. This way, by comparing 12 metrics, just taking into account word and multi-word based document representation, we could conclude that Tf-Idf and Phi Square based metrics enabled better precision and recall than equivalent precision/recall obtained by Rvar and MI based metrics that tend to extract rare terms. This contradicts results obtained by [1,2].

As we wanted to extend our methodology to morphologically rich languages, we introduced another document representation in terms of word prefixes and in that way corroborated the conclusions made by [3] in their work, where *Bubbled* variants showed interesting results for morphologically rich languages tested, especially for Portuguese.

This other representation led us to the usage of 8 metrics based on the same 4 kernel metrics already mentioned. Experiments were made for Portuguese, English and Czech. Higher precision obtained for Portuguese was obtained using two of the metrics designed to handle prefix, word and multi-word representation. For Czech, and even for English, results were not that spectacular but deserve further attention. As a matter of fact, second best precision for the 5 top ranked key terms candidates, both for Czech and for English was obtained by using Least Bubble Tf-Idf metric.

Again, Tf-Idf and Phi Square derived metrics were the best performing. Also, it is worth to mention that the Bubble operator enabled some improvements in the results obtained when applied to Rvar and MI metrics. It is worth noticing that, for Portuguese and Czech, for some metrics, precision augmented when we considered top 10 and even top 20 ranked extracted terms in relation to top 5 ranked ones. For Czech that occurred for Least Bubbled Median Tf-Idf and Least Bubbled Median Phi-Square. For Portuguese it was the case for Least Tf-Idf and Least Bubbled Phi-Square.

In future work we will mainly explore Tf-Idf and Phi-square metrics and their de rivatives. Then we must take a greater care of the length of texts evaluated. As a matter of fact, for a large text it may make sense an evaluation with 5, 10 or 20 best ranked terms. But for smaller texts taking just the 5 best ranked terms may affect negatively the mean precision of all documents as, in such cases, at most 2 or 3 best ranked terms will probably exhaust good possibilities for document content descriptors.

In what concerns human evaluation we will make an effort for better preparing this work phase in order to overcome evaluation disagreement by discussing the criteria to be used by evaluators while making them explicit.

Regarding the problem identified in section 5 related to having multi-words that are not independent, we must take greater care on this problem, knowing that it is not that easy to solve. Take another example of extracted good descriptors using Phi-Square metric from document 32006D0688 (*in http://eur-lex.europa.eu/en/legis/latest/chap1620.htm*). Below are the terms classified as good:

- asylum
- asylum and immigration
- immigration
- areas of asylum and immigration
- areas of asylum
- national asylum

If we filter out multi-words that are sub multi-words of larger multi-words., in the example above we would have got rid of “asylum and immigration” and “areas of asylum”. But as you see other filtering possibilities might be used. So this must be cautiously addressed. As a matter of fact we are not so sure that a long key term (5-gram) as “areas of asylum and immigration” is a better descriptor than “asylum and immigration”. Equivalently, it might be extrapolated for the example shown in Table 1, that multiword “group on ethics in science and new technologies”, that might be recaptured by binding top ranked multi-words having identical extremities is possibly a good descriptor. But again some care must be taken. If we want to directly extract longer multi-words as that “group on ethics in science and new technologies” we just need to fix the maximum multiword length, this has computational cost. For this work it was fixed at 5.

Concerning Czech, a stricter evaluation would not accept some of the terms that were taken as good as they were case marked and should not be. This will certainly require some language dependent tool filtering. That is more complex than simple lemmatization of words.

In future work, instead of using fixed length character prefixes of words we will pre-process our documents collection to automatically extract real word radicals using some of the existing language independent morphology learners like Linguistica [23] and Morfessor [24].

For Asian languages as Chinese or Japanese, we will apply the extractor [18,17] to characters instead of words and extract multi-character, 2-grams and 3-grams, and use single and multi-character strings ranked using the metrics proposed.

For German, the use of language independent morphology learners mentioned above, together with words and multi-words extracted the same way as we did for Portuguese, Czech or English will enable us to extend our methodology to a larger set of languages.

Acknowledgements. This was supported by the Portuguese Foundation for Science and Technology (FCT/MCTES) through funded research projects ISTRION (ref. PTDC/EIA-EIA/114521/2009).

References

1. da Silva, J.F., Lopes, G.P.: A Document Descriptor Extractor Based on Relevant Expressions. In: Lopes, L.S., Lau, N., Mariano, P., Rocha, L.M. (eds.) EPIA 2009. LNCS (LNAI), vol. 5816, pp. 646–657. Springer, Heidelberg (2009)
2. da Silva, J.F., Lopes, G.P.: Towards Automatic Building of Document Keywords. In: COLING 2010 - The 23rd International Conference on Computational Linguistics, Poster Volume, Pequim, pp. 1149–1157 (2010)
3. Teixeira, L., Lopes, G., Ribeiro, R.A.: Automatic Extraction of Document Topics. In: Camarinha-Matos, L.M. (ed.) DoCEIS 2011. IFIP AICT, vol. 349, pp. 101–108. Springer, Heidelberg (2011)
4. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys 34(1), 1–47 (2002)
5. da Silva, J.F., Lopes, G.P.: A Local Maxima Method and a Fair Dispersion Normalization for Extracting Multiword Units. In: Proceedings of the 6th Meeting on the Mathematics of Language, Orlando, pp. 369–381 (1999)
6. Jacquemin, C.: Spotting and discovering terms through natural language processing. MIT Press (2001)
7. Hulth, A.: Improved Automatic Keyword Extraction Given More Linguistic Knowledge. In: EMNLP 2003 Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 216–223. Association for Computational Linguistics, Stroudsburg (2003)
8. Ngonga Ngomo, A.-C.: Knowledge-Free Discovery of Domain-Specific Multiword Units. In: Proceedings of the 2008 ACM Symposium on Applied Computing, SAC 2008, pp. 1561–1565. ACM, Fortaleza (2008), doi:`http://doi.acm.org/10.1145/1363686.1364053`
9. Martínez-Fernández, J.L., García-Serrano, A., Martínez, P., Villena, J.: Automatic Keyword Extraction for News Finder. In: Nürnberger, A., Detyniecki, M. (eds.) AMR 2003. LNCS, vol. 3094, pp. 99–119. Springer, Heidelberg (2004)

10. Cigarrán, J.M., Peñas, A., Gonzalo, J., Verdejo, F.: Automatic Selection of Noun Phrases as Document Descriptors in an FCA-Based Information Retrieval System. In: Ganter, B., Godin, R. (eds.) ICFCA 2005. LNCS (LNAI), vol. 3403, pp. 49–63. Springer, Heidelberg (2005)
11. Liu, F., Pennell, D., Liu, F., Liu, Y.: Unsupervised Approaches for Automatic Keyword Extraction Using Meeting Transcripts. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL, pp. 620–628. Association for Computational Linguistics, Boulder (2009)
12. Katja, H., Manos, T., Edgar, M., Maarten, de R.: The impact of document structure on keyphrase extraction. In: Proceeding of the 18th ACM Conference on Information and Knowledge Management, pp. 1725–1728. ACM, Hong Kong (2009)
13. Mihalcea, R., Tarau, P.: TextRank: Bringing Order into Texts. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, pp. 404–411 (2004)
14. Turney, P.D.: Learning Algorithms for Keyphrase Extraction. Inf. Retr. 2(4), 303–336 (2000), doi:10.1023/a:1009976227802
15. Lemnitzer, L., Monachesi, P.: Extraction and evaluation of keywords from Learning Objects - a multilingual approach. In: Proceedings of the Language Resources and Evaluation Conference (2008)
16. Matsuo, Y., Ishizuka, M.: Keyword Extraction from a single Document using word Co-Occurence Statistical Information. International Journal on Articial Intelligence Tools 13(1), 157–169 (2004)
17. da Silva, J. F., Dias, G., Guilloré, S., Lopes, J.G. P.: Using LocalMaxs Algorithm for the Extraction of Contiguous and Non-contiguous Multiword Lexical Units. In: Barahona, P., Alferes, J.J. (eds.) EPIA 1999. LNCS (LNAI), vol. 1695, pp. 113–132. Springer, Heidelberg (1999)
18. Gomes, L.: Multi-Word Extractor (2009), `http://hlt.di.fct.unl.pt/luis/multiwords/index.html`
19. Douglas McIlroy, M.: Suffix arrays (2007), `http://www.cs.dartmouth.edu/~doug/sarray/`
20. Yamamoto, M., Church, K.W.: Using Suffix Arrays to Compute Term Frequency and Document Frequency for All Substrings in a Corpus. Computational Linguistics 27(1), 1–30 (2001)
21. Everitt, B.S.: The Cambridge Dictionary of Statistics, 2nd edn. Cambridge University Press, New York (2002)
22. Manning, C.D., Raghavan, P., Schütze, H.: An Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
23. Goldsmith, J.: Unsupervised learning of the morphology of a natural language. Computational Linguistiscs 27(2), 153–198 (2001)
24. Creutz, M., Lagus, K.: Unsupervised models for morpheme segmentation and morphology learning. ACM Trans. Speech Lang. Process. 4(1), 1–34 (2007)

Action-Driven Perception for a Humanoid

Jens Kleesiek[1,3], Stephanie Badde[2], Stefan Wermter[3], and Andreas K. Engel[1]

[1] Department of Neurophysiology and Pathophysiology,
University Medical Center Hamburg-Eppendorf, Hamburg, Germany
[2] Department of Biological Psychology and Neuropsychology,
University of Hamburg, Hamburg, Germany
[3] Department of Informatics, Knowledge Technology,
University of Hamburg, Hamburg, Germany
{j.kleesiek,ak.engel}@uke.uni-hamburg.de,
Stephanie.Badde@uni-hamburg.de,
wermter@informatik.uni-hamburg.de

Abstract. We present active object categorization experiments with a real humanoid robot. For this purpose, the training algorithm of a recurrent neural network with parametric bias has been extended with adaptive learning rates. This modification leads to an increase in training speed. Using this new training algorithm we conducted three experiments aiming at object categorization. While holding different objects in its hand, the robot executes a motor sequence that induces multi-modal sensory changes. During learning, these high-dimensional perceptions are 'engraved' in the network. Simultaneously, low-dimensional PB values emerge unsupervised. The geometrical relation of these PB vectors can then be exploited to infer relations between the original high dimensional time series characterizing different objects. Even sensations belonging to unknown objects can be discriminated from known (learned) ones and kept apart from each other reliably. Additionally, we show that the network tolerates noisy sensory signals very well.

Keywords: Active Perception, RNNPB, Humanoid Robot.

1 Introduction

Motor actions determine the sensory information that agents receive from their environment. Combining sensory and motor processes dynamically facilitates many tasks, one of those being object classification.

The intention of this experiment is to provide a neuroscientifically and philosophically inspired model for *what do objects feel like?* For this purpose, we stress the active nature of perception within and across modalities. According to the sensorimotor contingencies theory [1], actions are fundamental for perception and help to distinguish the qualities of sensory experiences in different sensory channels (e.g. 'seeing' or 'touching'). O'Regan and Noë actually suggest that "seeing is a way of acting" [1]. Exactly this is mimicked in our computational study.

It has been shown that if the fruit fly drosophila cannot recognize a pattern it starts to move [2]. It is also known that flies use motion to visually determine the depth of

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 83–99, 2013.

perceived obstacles [3]. Similarly, pigeons bob their heads up and down to recover depth information [4]. Not only living beings, but robots too are embodied [5], and they have the ability to act and to perceive. In the presented experiments the robot actually needs to act to perceive the objects it holds in its hand. The action-driven sensations are guided by the physical properties of its body, the world and the interplay of both.

A humanoid robot moves toy bricks up and down and rotates them back and forth, while holding them in its hand. The induced multi-modal sensory impressions are used to train a modified version of a recurrent neural network with parametric bias (RNNPB), originally developed by Tani and Ito [6]. The robot is able to self-organize the contextual information and in turn, to use this learned sensorimotor knowledge for object classification. Due to the overwhelming generalization capabilities of the recurrent architecture, the robot is even able to correctly classify unknown objects. Furthermore, we show that the proposed model is very robust against noise.

2 Theory

Despite its intriguing properties, the recurrent neural network with parametric bias has hardly been used by anybody other than the original authors. Mostly, the architecture is utilized to model the mirror neuron system [7,8]. Here we apply the variant proposed by Cuijpers *et al.* [8] using an Elman-type structure [9] at its core. Furthermore, we modify the training algorithm to include adaptive learning rates for training of the weights, as well as the PB values. This results in an architecture that is more stable and converges faster.

2.1 Storage

The recurrent neural network with parametric bias (an overview of the architecture unfolded in time can be seen in Fig. 1) can be used for the storage, retrieval and recognition of sequences. For this purpose, the parametric bias (PB) vector is learned simultaneously and *unsupervised* during normal training of the network. The prediction error with respect to the desired output is determined and backpropagated through time using the BPTT algorithm [9]. However, the error is not only used to correct all the synaptic weights present in the Elman-type network. Additionally, the error with respect to the PB nodes δ^{PB} is accumulated over time and used for updating the PB values after an entire forward-backward pass of a single time series, denoted as epoch e. In contrast to the synaptic weights that are shared by all training patterns, a unique PB vector is assigned to each individual training sequence. The update equations for the i-th unit of the parametric bias pb for a time series of length T is given as:

$$\rho_i(e+1) = \rho_i(e) + \gamma_i \sum_{t=1}^{T} \delta_{i,t}^{\mathrm{PB}} , \tag{1}$$

$$pb_i(e) = \mathrm{sigmoid}(\rho_i(e)) , \tag{2}$$

where γ is the update rate for the PB values, which in contrast to the original version is not constant during training and not identical for every PB unit. Instead, it is scaled

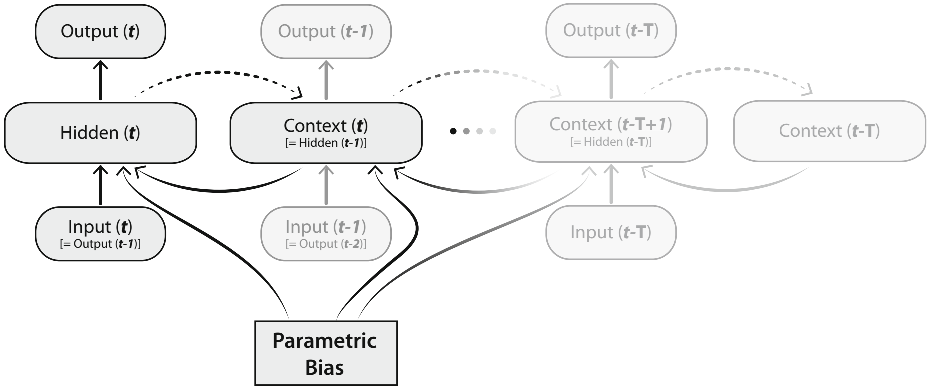


Fig. 1. Network architecture. The Elman-type Recurrent Neural Network with Parametric Bias (RNNPB) unfolded in time. Dashed arrows indicate a verbatim copy of the activations (weight connections set equal to 1.0). All other adjacent layers are fully connected. t is the current time step, T denotes the length of the time series.

proportionally to the absolute mean value of prediction errors being backpropagated to the i-th node over time T:

$$\gamma_i \propto \frac{1}{T} \left\| \sum_{t=1}^{T} \delta_{i,t}^{\mathrm{PB}} \right\| \quad . \tag{3}$$

The other adjustable weights of the network are updated via an adaptive mechanism, inspired by the resilient propagation algorithm proposed by Riedmiller and Braun [10]. However, there are decisive differences. First, the learning rate of each neuron is adjusted after every epoch. Second, not the sign of the partial derivative of the corresponding weight is used for changing its value, but instead the partial derivative itself is taken.

To determine if the partial derivative of weight w_{ij} changes its sign we can compute:

$$\epsilon_{ij} = \frac{\partial E_{ij}}{\partial w_{ij}}(t-1) \cdot \frac{\partial E_{ij}}{\partial w_{ij}}(t) \tag{4}$$

If $\epsilon_{ij} < 0$, the last update was too big and the local minimum has been missed. Therefore, the learning rate η_{ij} has to be decreased by a factor $\xi^- < 1$. On the other hand, a positive derivative indicates that the learning rate can be increased by a factor $\xi^+ > 1$ to speed up convergence. This update of the learning rate can be formalized as:

$$\eta_{ij}(t) = \begin{cases} \mathbf{max}(\eta_{ij}(t-1) \cdot \xi^-, \eta_{min}) & \text{if } \epsilon_{ij} < 0, \\ \mathbf{min}(\eta_{ij}(t-1) \cdot \xi^+, \eta_{max}) & \text{if } \epsilon_{ij} > 0, \\ \eta_{ij}(t-1) & \text{else.} \end{cases} \tag{5}$$

The succeeding weight update Δw_{ij} then obeys the following rule:

$$\Delta w_{ij}(t) = \begin{cases} -\Delta w_{ij}(t-1) & \text{if } \epsilon_{ij} < 0, \\ \eta_{ij}(t) \cdot \frac{\partial E_{ij}}{\partial w_{ij}}(t) & \text{else.} \end{cases} \tag{6}$$

In addition to reverting the previous weight change in the case of $\epsilon_{ij} < 0$ the partial derivative is also set to zero ($\frac{\partial E_{ij}}{\partial w_{ij}}(t) = 0$). This prevents changing of the sign of the derivative once again in the succeeding step and thus a potential double punishment.

We use a nonlinear activation function with parameters recommended by LeCun *et al.* [11] for all neurons in the network, as well as for the PB units (Eq. 2):

$$\mathrm{sigmoid}(x) = 1.7159 \cdot \tanh\left(\frac{2}{3} \cdot x\right) . \tag{7}$$

2.2 Retrieval

The PB vector is usually low dimensional and resembles bifurcation parameters of a nonlinear dynamical system, i.e. it characterizes fixed-point dynamics of the RNN. During training the PB values are self-organized, thereby encoding each time series and arranging it in PB space according to the properties of the training pattern. This means that the values of similar sequences are clustered together, whereas more distinguishable ones are located further apart. Once learned, the PB values can be used for the generation of the time series previously stored. For this purpose, the network is operated in closed-loop mode. The PB values are 'clamped' to a previously learned value and the forward pass of the network is executed from an initial input $I(0)$. In the next time steps, the output at time t serves as an input at time $t + 1$. This leads to a reconstruction of the training sequence with a very high accuracy (limited by the convergence threshold used during learning).

2.3 Recognition

A previously stored (time) sequence can also be recognized via its corresponding PB value. Therefore, the observed sequence is fed into the network without updating any connection weights. Only the PB values are accumulated according to Eq. 1 and 2 using a constant learning rate γ this time. Once a stable PB vector is reached, it can be compared to the one obtained during training.

2.4 Generalized Recognition and Generation

The network has substantial generalization potential. Not only previously stored sequences can be reconstructed and recognized. But, (time) sequences apart from the stored patterns can be generated. Since only the PB values but not the synaptic weights are updated in recognition mode, a stable PB value can also be assigned to an unknown sequence.

For instance, training the network with two sine waves of different frequencies allows cyclic functions with intermediate frequencies to be generated simply by operating the network in generation mode and varying the PB values within the interval of the PB values obtained during training. Furthermore, the PB values obtained during recognition of a previously unseen sine function with an intermediate frequency (w.r.t. the training sequences) will lie within the range of the PB values acquired during learning. Hence,

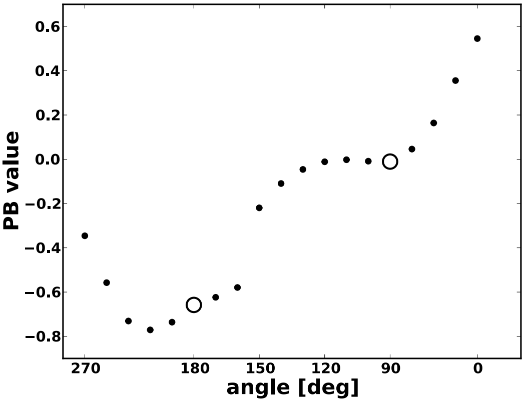


Fig. 2. Generalized recognition of trained and untrained sequences. The PB values of the two trained 2-D time series using Eq. 8 with $\theta = 90$ and $\theta = 180$, respectively, are marked using white circles. In contrast, the PB values obtained by feeding the network with untrained sequences generated with varying θ values are drawn as black dots. These values are arranged in a structured way, emphasizing the self-organization of the PB space.

the network is able to capture a reciprocal relationship between a time series and its associated PB value.

These generalized recognition and generation capabilities of the modified RNNPB are demonstrated in a more complex example. For this purpose, consider the 2-D sinusoidal sequences described by the following equation:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 0.5 \begin{pmatrix} cos\ \theta & -sin\ \theta \\ sin\ \theta & cos\ \theta \end{pmatrix} \begin{pmatrix} sin\ \frac{\pi \cdot t}{6} \\ cos\ \frac{\pi \cdot t}{12} \end{pmatrix} \tag{8}$$

Plotting x_1 *vs.* x_2 results in a figure-eight shape that is rotated according to the angular value specified by θ. Two 2-D time series of length $t = 25$ were generated using $\theta = 90$ and $\theta = 180$, respectively. In contrast to the robot experiments presented below, the network only has a single PB unit.

After training, the network is able to recognize those sequences based on their trained PB values ($\text{PB}_{\theta:90} = -0.01107$ and $\text{PB}_{\theta:180} = -0.65604$), which differ only by a small amount ($\epsilon_{\theta:90} = 0.0005$ and $\epsilon_{\theta:180} = 0.002$) from the ones obtained during storage. The PB values of the two trained sequences are plotted in Fig. 2 using white circular markers. Next to the trained sequences, the network is also fed with novel, previously untrained, sequences. These are generated using Eq. 8 with varying θ values. The network also generates stable PB values for those unknown sequences (black dots in Fig. 2). It can be seen that the PB values are ordered according to the angular value of the underlying time series. This reciprocal relationship can be used to infer the angular value of a sequence generated with an unknown θ value. Thus, varying the PB value results in a rotation of the figure-eight shape.

2.5 Evaluation of the Adaptive Learning Rate

To evaluate the adaptive learning rate proposed in Sec. 2.1, artificial 1-D test data of length $T = 11$ in the interval $[-\pi; \pi[$ is generated using the following equations:

$$x = \sin(t)\,, \tag{9}$$

$$x = \frac{\sin(3t) \cdot \sin(t)}{2t^2} - 0.5\,. \tag{10}$$

Eq. 9 is referred to as `sin` and Eq. 10 as `sinc`. Except for the following differences, the RNNPB network parameters were identical to the parameters of the robot experiments (see below). The architecture contained only one input and one output node, as well as only one PB unit. The convergence criterion was set to 10^{-4}.

2.6 Network Parameters for Robot Experiments

Based on systematic empirical trials, the following parameters have been determined for our experiments. The network contained two input and two output nodes, 24 hidden and 24 context neurons as well as 2 PB units. The convergence criterion for back propagation through time (BPTT) was set to 10^{-6} in the first, and 10^{-5} in the second experiment. For recognition of a sequence, the update rate γ of the PB values was set to 0.1. The values for all other individual adaptive learning rates (Eq. 5) during training of the synaptic weights were allowed to be in the range of $\eta_{min} = 10^{-12}$ and $\eta_{max} = 50$; depending on the gradient they were either increased by $\xi^{+} = 1.01$ or decreased by a factor $\xi^{-} = 0.9$.

3 Scenario

The humanoid robot Nao[1] is programmed to conduct the experiments (Fig. 3 a). The task for the robot is to identify which object (toy brick) it holds in its hand. In total there are eight object categories that have to be distinguished by the robot: the toy bricks have four different shapes (circular-, star-, rectangular- and triangular-shaped), of which each exists in two different weight versions (light and heavy). Hence, for achieving a successful classification multi-modal sensory impressions are required. Additionally, *active* perception is necessary to induce sensory changes essential for discrimination of –depending on the perspective– similar looking shapes (e.g. star- and circular-shaped objects). For this purpose, the robot performs a predefined motor sequence and simultaneously acquires visual and proprioceptive sensor values.

3.1 Data Acquisition

The recorded time series comprises 14 sensor values for each modality. In each single trial the robot turns its wrist with the object between its fingers by 45.8 ° back and forth

[1] `http://www.aldebaran-robotics.com`

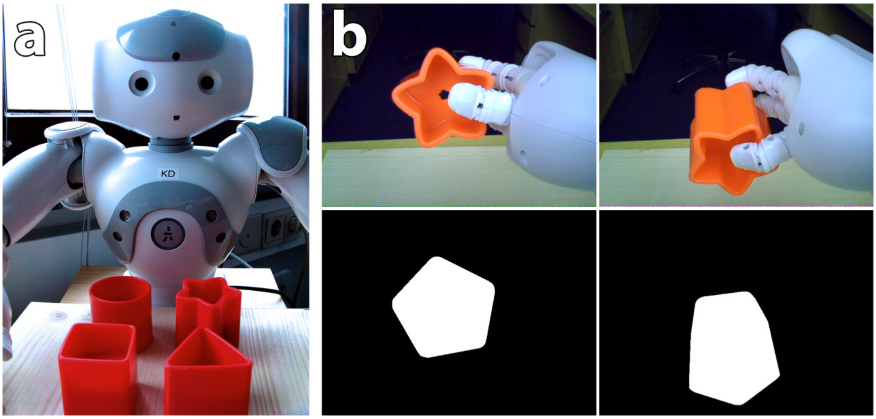


Fig. 3. Scenario. a) Toy bricks in front of the humanoid robot Nao. The toy bricks exist in four different shapes, have an identical color and are either light-weight ($15\,g$) or heavy ($50\,g$). This results in a total of eight categories that have to be distinguished by the robot. b) Rotation movement with the star-shaped object captured by the robot camera. In the upper row the raw camera image is shown, whereas the bottom row depicts the preprocessed image that is used to compute the visual features.

twice, followed by lifting the object up and down three times (thereby altering the pitch of the shoulder joint by $11.5\,°$) and, finally, turning it again twice.

After an action has been completed, the raw image of the lower camera of the Nao robot is captured, whereas the electric current of the shoulder pitch servo motor is recorded constantly (sampling frequency $10\,Hz$) over the entire movement interval. For each object category 10 single trial time series are recorded in the described way and processed in real-time. This yields 80 bi-modal time series in total.

3.2 Data Processing

For the proprioceptive measurements only the mean values are computed for the time intervals in between movements. The visual processing, on the other hand, involves several steps (Fig. 3 b), which are accomplished using OpenCV [12]. First, the raw color image is converted to a binary image using a color threshold. Next, the convex hull is computed and, based on that, the contour belonging to the toy brick is extracted [13]. For the identified contour the first Hu moment is calculated [14]. Finally, the visual measurements are scaled to be within the interval $[-0.5, 0.5]$.

3.3 Training and Test Data

For testing, the data of single trials are used, i.e. 10 2-D time series per object category (one dimension for each modality). However, for training, a prototype is determined for each object category and modality (Fig. 4). To obtain this subclass representative, the mean value of pooled single trials, with regard to identical object properties, is computed. This means that, for instance, all circular-shaped objects are combined ($n = 20$)

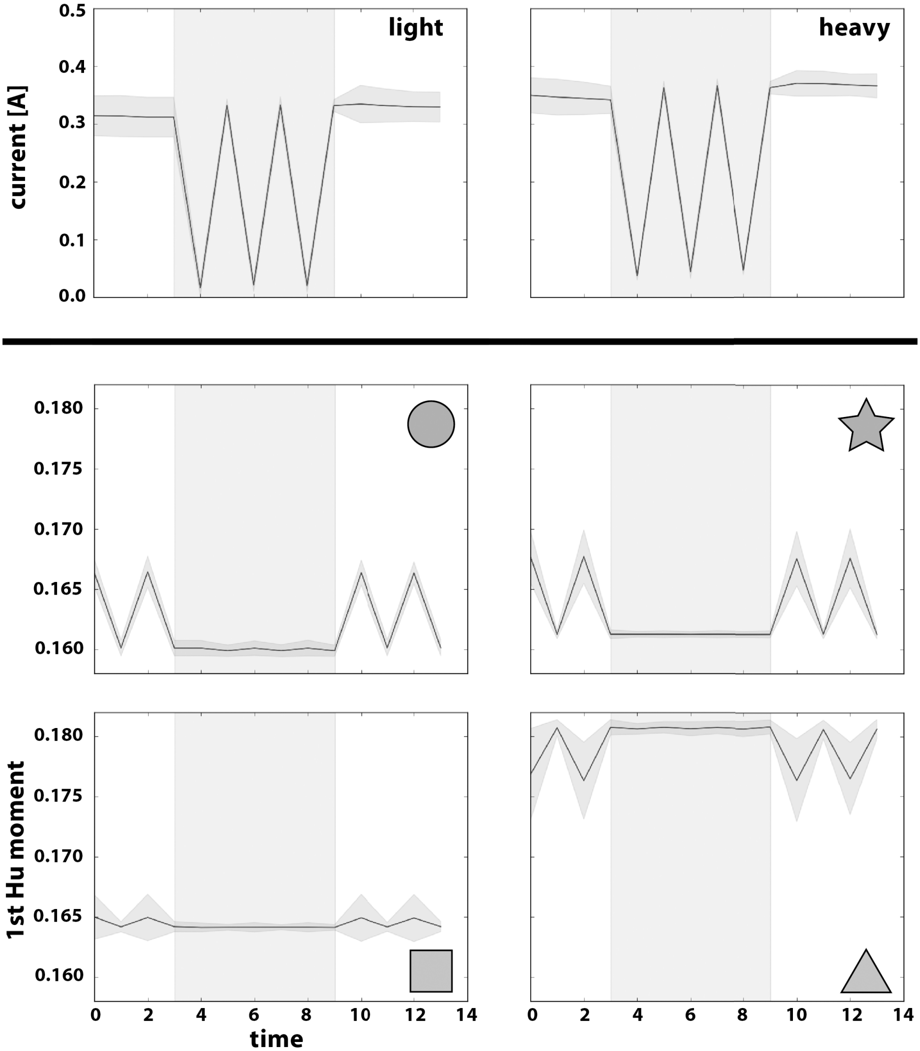


Fig. 4. Training data. The mean values of the two weight conditions (light and heavy, top) and the four visual conditions (matching symbols, bottom) are shown. These mean time series are used as prototypes for training the RNNPB. Vertical gray shaded areas represents the up and down movement, whereas back and forth movements are unshaded. The area surrounding the signals delineates two standard deviations from the mean.

and used to compute the visual prototype for circular-shaped objects. To find the proprioceptive prototype for e.g. all heavy objects, all individual measurements with this property ($n = 40$) are aggregated and used to calculate the mean value at each time step. The subclass prototypes are then combined to form a 2-D multi-modal time series that serves as an input for the recurrent neural network during training.

4 Results

4.1 Evaluation of the Adaptive Learning Rate

To evaluate the improvements caused by introduction of the adaptive learning rate as described in Sec. 2.1, an RNNPB was trained 1000 times with two 1-D sequences (Eq. 9 and 10). The results are statistically evaluated using a t-test. To compensate for the sample size bias, the optimal sample size was determined based on the mean value and the standard deviation of the data. This optimal sample size was used to draw 10,000 random subsets of the data, which were subsequently evaluated to obtain an average p-value for the t-test. The results are summarized in Tab. 1. The modifications lead, on average, to a 22-fold speedup of the training times ($t(5) = -17.13, p = 0.000$) for this particular example. Also the number of recognition steps has improved significantly ($t(20) = -3.55, p = 0.002$). However, no significant changes of the retrieval accuracy measured with the mean squared error (MSE) can be found.

Table 1. Statistical evaluation of the adaptive learning rate. Mean values and standard deviations are shown, significant changes (t-test, $p < 0.005$) are marked bold.

	Modified RNNPB	Classical RNNPB	Factor
Total steps	**5,520 (±1,713)**	122,709 (±20,027)	22.2
Total time	**34** s **(±10)**	751 s (±124)	22
MSE *sin*	4.3×10^{-4} ($\pm 1.2 \times 10^{-3}$)	5.5×10^{-4} ($\pm 3 \times 10^{-4}$)	–
MSE *sinc*	4.7×10^{-4} ($\pm 8.7 \times 10^{-4}$)	1.9×10^{-4} ($\pm 1.9 \times 10^{-4}$)	–
Recognition steps	**192 (±85)**	284 (±101)	1.48

Plotting the average MSE against the number of steps needed until the convergence criterion is reached, further highlights the drastic improvement in speed (Fig. 5). The error, shown separately for both sequences, decreases for both algorithms in a similar manner. However, the adaptive version looks 'compressed' in comparison to the classical algorithm. In addition, the fluctuations are reduced, indicating a more stable behavior of the modified RNNPB.

4.2 Classification Using All Object Categories for Training

In the first experiment the modified recurrent neural network with parametric bias was trained with the bi-modal prototype time series of all eight object categories (Sec. 3.3). During training, the PB values for the respective categories emerged in an unsupervised way. This means, the two-dimensional PB space self-organizes based on the inherent properties of the sensory data that was presented to the network. Hence, objects with similar dynamic sensory properties are clustered together. This can be seen in Fig. 6. For instance, the learned PB vectors representing star- and circular-shaped objects, either light-weight (white symbol) or heavy (black symbol), are located in close proximity,

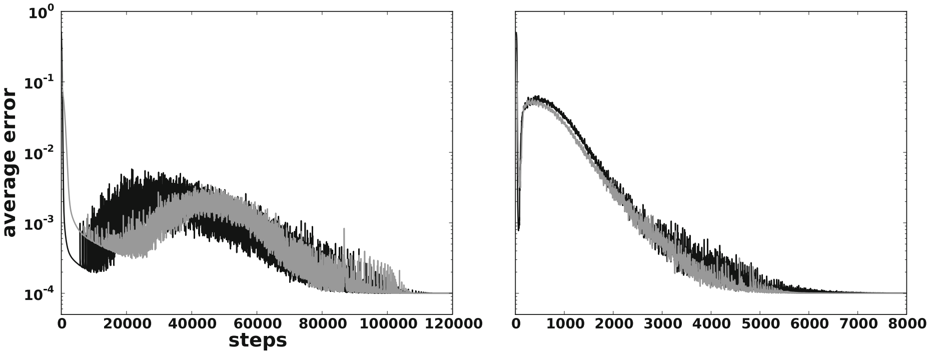


Fig. 5. Error plot comparing classical (left) to modified (right) RNNPB. The mean squared error (MSE) of the `sin` sequence is shown in black, whereas the average MSE of the `sinc` sequence is drawn in gray.

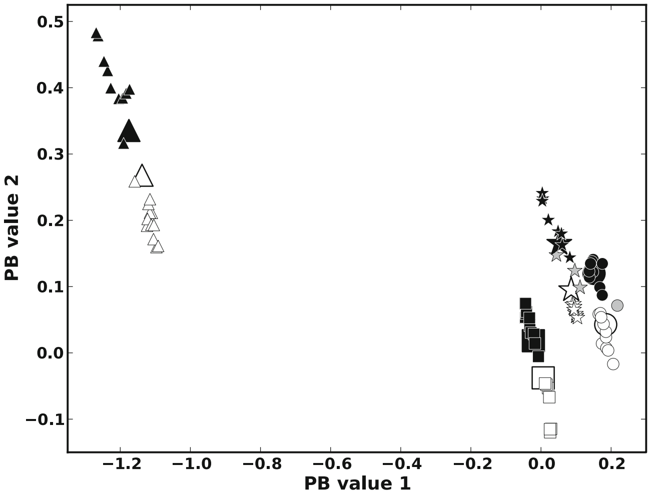


Fig. 6. Classification using all object categories for training. PB values are depicted using white (light-weight) and black (heavy) symbols matching the corresponding shape. Large symbols represent the class prototypes used for training. Smaller symbols depict PB values obtained during testing with bi-modal single trial data. If the objects have not been correctly classified, they are shown gray.

whereas the PB values coding for the triangular-shaped objects are positioned more distantly. This is due to the deviating visual sensory impression they generate (Fig. 4). The experiment has been repeated several times with different random initializations of the network weights. However, the obtained PB values of the different classes always demonstrate a comparable geometric relation with respect to each other.

To demonstrate the retrieval properties (Sec. 2.2) of the fully trained architecture, the PB values acquired during training were ‘clamped’ to the network. Operating the network in closed-loop mode showed that the input sequences used for training can be

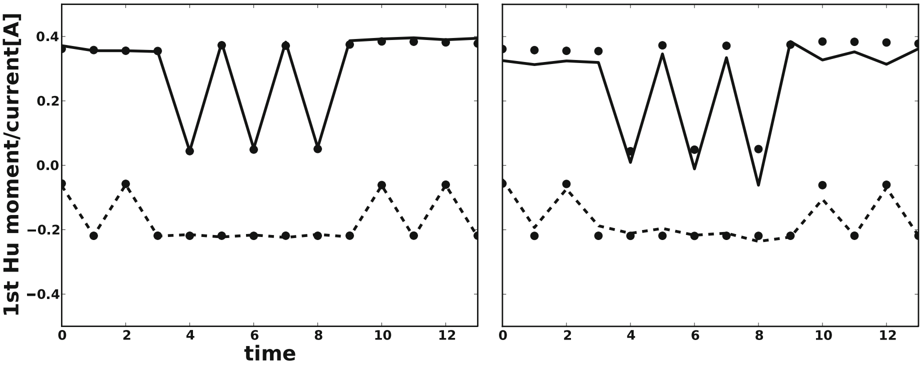


Fig. 7. Retrieval and generation capabilities. Solid dots represent the proprioceptive and visual sampling points of the heavy star-shaped prototype time series (Fig. 4). Dashed (visual) lines and solid (proprioceptive) lines are the time series generated by the network operated in closed-loop with 'clamped' PB values as the only input. The PB values have been acquired unsupervised either during full training (left) or partial training (right). During partial training (right) the network has only been trained with the prototype sequences for the light-weight circle and the heavy triangle. Still, the network is able to generate a fairly accurate sensory prediction for the (untrained) heavy star-shaped object.

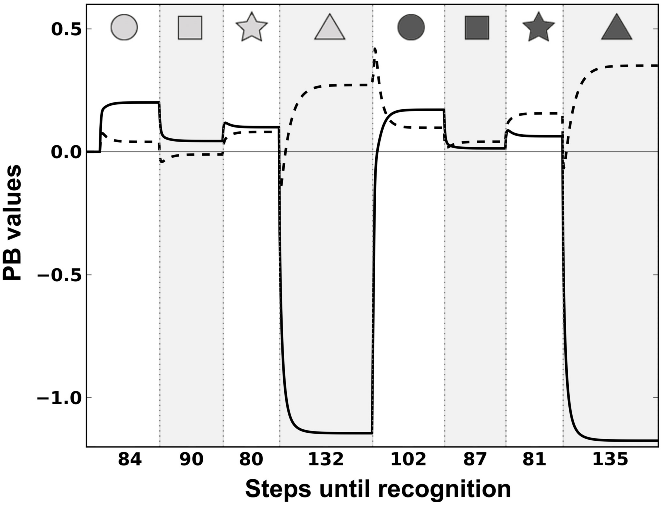


Fig. 8. Steps until stable PB values are reached. Bi-modal sensory sequences for all light-weight and heavy objects (represented by matching symbols in light and dark gray, respectively) are consecutively fed into the network. The time courses of PB value 1 (solid line) and PB value 2 (dashed line) during the recognition process are plotted.

retrieved with a very high accuracy. As an example this is shown in Fig. 7 (left) for the heavy star-shaped object.

The steps needed until stable PB values are reached, which in turn can be used for recognition, are illustrated in Fig. 8. The bi-modal sensory sequences for all light-weight and heavy objects were fed consecutively into the network. On average it took less than 100 steps (about 200 *ms* on a contemporary desktop computer) until the PB values converged. The convergence criterion was set to 20 consecutive iterations where the cumulative change of both PB values was $< 10^{-5}$. To assure that the PB values reached a stable state, this number has been successfully increased to 100,000 consecutive steps in preliminary experiments (not shown). Note, that the network and PB values was not reinitialized when the next sensory sequence was presented to the network. Thus, the robot can continuously interact with the toy bricks and is able to immediately recognize an object based on its sensorimotor sequence.

For testing, the network was operated in generalized recognition mode (Sec. 2.4). Single trial bi-modal sensory sequences were presented to the network that in turn provided an 'identifying' PB value. The class membership, i.e. which object the robot holds in its hand and how heavy this object is, was then determined based on the minimal Euclidean distance to the PB values of the class prototypes (large symbols). In Fig. 6 the PB values of all 80 single trial test patterns are depicted.

Only 4 out of 80 objects are misclassified (shown in gray), yielding an error rate of 5 %. Interestingly, only star- and circular-shaped objects are confused by the network, which indeed generate very similar sensory impressions (cf. Fig. 4). To assess the meaning of the error rate and estimate how challenging the posed problem is, we evaluated the data with two other commonly used techniques in machine learning. First, we trained a multi-layer perceptron (28 input, 14 hidden and one output unit) with the prototype sequences. Testing with the single trial data resulted in an error rate of 46.8 %, reflecting weaker generalization capabilities of the non-recurrent architecture. Next, we trained and evaluated our data with a support vector classifier (SVC) using default parameters [15]. In contrast, this method is able to classify the data perfectly.

4.3 Classification Using Only the Light Circular-Shaped and the Heavy Triangular-Shaped Object for Training

In the second experiment, only the bi-modal prototypes for the light circular- and heavy triangular-shaped objects were used to train the RNNPB. Although, the absolute PB values obtained during training differ from the ones being determined in the previous experiment, their relative Euclidean distance in PB space is nearly the same (1.39 *vs.* 1.35), stressing the data-driven self-organization of the parametric bias space.

For testing, initially only the bi-modal sensory time series matching the two training conditions were fed into the network, thereby determining their PB values. Using the Euclidean distance subsequently to obtain the class membership resulted in a flawless identification of the two categories.

Further evaluation of the single trial test data was performed in two stages. In a primary step the remaining test data was presented to the network and the respective PB values were computed (generalized recognition, Sec. 2.4). Despite not having been trained with prototypes for the remaining six object categories, the network is able to cluster PB values stemming from similar sensory situations, i.e. identical object categories. In a succeeding step we computed the centroid for each class (mean PB value) and classified again based on the Euclidean distance. This time only two single trial time series were misclassified by the network (error rate 2.5 %). The results are shown in Fig. 9.

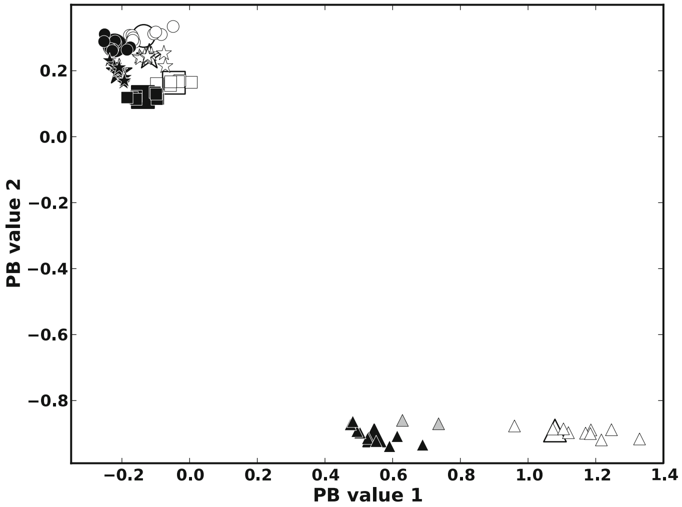


Fig. 9. Classification using only the light circular-shaped and the heavy triangular-shaped object for training. PB values of the class prototypes and the a posteriori computed cluster centers of the untrained object categories are depicted using larger symbols that match the objects shape. Smaller symbols are used for PB values of sensory data of single trials. If the objects have not been classified correctly they are shown in gray, otherwise white is used for light-weight and black for heavy-weight objects.

The generalization potential (Sec. 2.4) of the architecture is presented in Fig. 7 (right) for the heavy star-shaped object. For this purpose, the mean PB values (centroid of the respective class) were clamped to the network, which was operated in closed-loop mode. The network had only been trained with the light circular- and the heavy triangular-shaped object. Still, it was possible to generate sensory predictions for unseen objects, e.g. the heavy star-shaped toy brick, that match the real sensory impressions fairly well.

4.4 Noise Tolerance

Based on the network weights that had been obtained in experiment 2 (training the RNNPB only with the bi-modal prototypes for the light circular- and heavy triangular-shaped objects), we evaluated the noise tolerance of the recurrent neural architecture. For this purpose, uniformly distributed noise of increasing levels was added to the visual prototype time series (Fig. 10).

Even high levels of noise allow for a reliable linear discrimination of the two classes. Furthermore, the PB values of increasing noise levels show commonalities and are clustered together, again providing evidence for a data-driven self-organization of the PB space. Thus, determining the Euclidean distance of the PB values obtained from the noisy signals to the class representatives enables not only the class membership to be determined, it also allows the noise level to be estimated with respect to the prototypical sensory impression. It also can be show that the network tolerates noise added to the time series of both modalities very well [16].

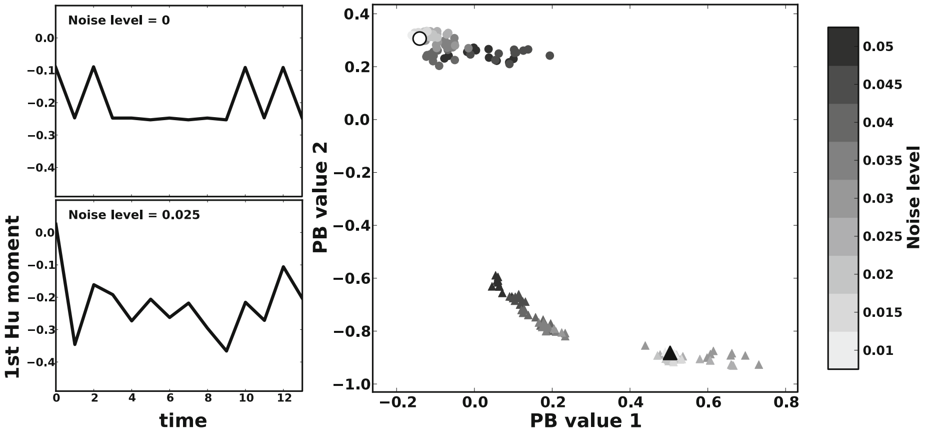


Fig. 10. Noise tolerance. On the right uniformly distributed noise of increasing levels (grayscale coded) is added to the visual prototype time series for the light-weight circle and the heavy triangle. The PB values are determined and marked with a matching symbol. The white circle and large black triangle show the PB values obtained during training without noise. On the left the impact of the noise level is shown exemplarily for the visual prototype time series of the circular shape.

5 Discussion

By introducing modifications to the learning algorithm of the RNNPB we were able to achieve a significant 22-fold increase in speed (Tab. 1) for the storage of two 1-D signals. It was also confirmed that the storage and retrieval of those time series was stable and that learning converged in a well-behaved manner (Fig. 5). Admittedly, the storage of other sequences with e.g. a different dimensionality, length or dynamic, may well result in a different performance outcome.

After confirming flawless operation of the training algorithm we conducted three experiments aiming at object categorization. While holding different objects (Sec. 3) in its hand, the robot executes a motor sequence that induces multi-modal sensory changes. During learning these high-dimensional perceptions are ‘engraved’ in the network. Simultaneously, low-dimensional PB values emerge unsupervised, coding for a sensorimotor sequence characterizing the interplay of the robot with an object. We show that 2-D time series of length $T = 14$ can be reliably represented by a 2-D PB vector and that this vector allows learned sensory sequences to be recalled with a high accuracy (Fig. 7 left). Furthermore, the geometrical relation of PB vectors of different objects can be used to infer relations between the original high dimensional time series, e. g. the sensation of a star-shaped object ‘feels’ more like a circular-shaped object than a triangular-shaped one. Due to the experimental noise of single trials, identical objects cause varying sensory impressions. Still, the RNNPB can be used to recognize those (Fig. 6). Additionally, sensations belonging to unknown objects can be discriminated from known (learned) ones. Moreover, sensations arising from different unknown objects can be kept apart from each other reliably (Fig. 9).

Comparing the classification results of the fully trained RNNPB with the SVC reveals a superior performance of the support vector classifier. Nevertheless, it has to be kept in mind that the maximum margin classifier cannot be used to generate or retrieve time series. Interestingly, the error rate is lower if the recurrent network is only trained with two object categories (Sec. 4.3). A potential explanation, besides random fluctuations, could be that during training a common set of weights has to be found for all object categories. This process presumably interferes, due to the challenging input data, with the self-organization of the PB space.

A drawback of the presented model is that it currently operates on a fixed motor sequence. It would be desirable if the robot performed *motor babbling* [17] leading not only to a self-organization of the sensory space, but to a self-organization of the sensorimotor space. A simple solution to this problem would be to train the network additionally with the motor sequence most appropriate for an object, i.e. reflecting its affordance [18]. This would lead to an even better classification result because the motor sequences themselves would help to distinguish the objects from each other and, thus, the emerging PB values would be arranged further apart in PB space (conversely, this means currently it does not make sense to train the network with the identical motor sequences in addition). However, that does not address the fact that the robot should identify the object affordances, the movements characterizing an object, by itself.

In related research, Ogata *et al.* also extract multi-modal dynamic features of objects, while a humanoid robot interacts with them [19]. However, there are distinct differences. Despite using fewer objects in total, the problem posed in our experiments is considerably harder. Our toy bricks have approximately the same circumference and identical color. Furthermore, they exist in two weight classes with an identical in-class weight that can only be discriminated via multi-modal sensory information. We provide classification results, compare the results to other methods (MLP and SVC) and evaluate the noise tolerance of the architecture. In addition, only prototype time series are used for training (in contrast to using all single-trial time series), resulting in a reduced training time. Further, it is demonstrated that, if the network has already acquired sensorimotor knowledge of certain objects, it is able to generalize and provide fairly accurate sensory predictions for unseen ones (Fig. 7 right).

There are several potential applications of the presented model. As shown in Fig. 10, the network tolerates noise very well. This fact can be exploited for sensor de-noising. Despite receiving a noisy sensory signal, the robot still will be able to determine the PB values of the class representative based on the Euclidean distance. In turn, these values can be used to operate the RNNPB in retrieval mode (Sec. 2.2), generating the noise-free sensory signal previously stored, which then can be processed further. In fact, Körding and Wolpert suggested that the central nervous system combines, in nearly optimal fashion, visual, proprioceptive and other sensory information to overcome sensory and motor noise [20]. Next to their Bayesian framework an RNNPB might also be a possible way to model this 'de-noising' happening in the brain.

In conclusion, we present a promising framework for object classification based on action-driven perception implemented on a humanoid robot. The underlying design principles are rooted in neuroscientific and philosophical hypotheses.

Acknowledgements. This work was supported by the Sino-German Research Training Group CINACS, DFG GRK 1247/1 and 1247/2, and by the EU projects KSERA under 2010-248085 and eSMCs under ICT-270212. We thank R. Cuijpers and C. Weber for inspiring and very helpful discussions, S. Heinrich, D. Jessen and N. Navarro for assistance with the robot.

References

1. O'Regan, J.K., Noë, A.: A sensorimotor account of vision and visual consciousness. Behav. Brain Sci. 24(5) (October 2001); 939–73; discussion 973–1031
2. Dill, M., Wolf, R., Heisenberg, M.: Visual pattern recognition in drosophila involves retinotopic matching. Nature 365(6448), 751–753 (1993)
3. Franceschini, N.: Combined optical, neuroanatomical, electrophysiological and behavioral studies on signal processing in the fly compound eye. In: Taddei-Ferretti, C. (ed.) Biocybernetics of Vision: Integrative Mechanisms and Cognitive Processes: Proceedings of the International School of Biocybernetics, Casamicciola, Napoli, Italy, October 16-22, 1994, vol. 2, World Scientific, Singapore (1997)
4. Steinman, S.B., Steinman, B.A., Garzia, R.P.: Foundations of binocular vision: a clinical perspective. McGraw-Hill, New York (2000)
5. Pfeifer, R., Lungarella, M., Iida, F.: Self-organization, embodiment, and biologically inspired robotics. Science 318(5853), 1088–1093 (2007)
6. Tani, J., Ito, M.: Self-organization of behavioral primitives as multiple attractor dynamics: A robot experiment. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 33(4), 481–488 (2003)
7. Tani, J., Ito, M., Sugita, Y.: Self-organization of distributedly represented multiple behavior schemata in a mirror system: reviews of robot experiments using RNNPB. Neural Netw. 17(8-9), 1273–1289 (2004)
8. Cuijpers, R.H., Stuijt, F., Sprinkhuizen-Kuyper, I.G.: Generalisation of action sequences in RNNPB networks with mirror properties. In: Proceedings of the 17th European symposium on Artifical Neural Networks (ESANN), pp. 251–256 (2009)
9. Kolen, J.F., Kremer, S.C.: A field guide to dynamical recurrent networks. IEEE Press, New York (2001)
10. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In: IEEE International Conference on Neural Networks, vol. 1, pp. 586–591 (1993)
11. LeCun, Y.A., Bottou, L., Orr, G.B., Müller, K.-R.: Efficient BackProp. In: Orr, G.B., Müller, K.-R. (eds.) NIPS-WS 1996. LNCS, vol. 1524, pp. 9–50. Springer, Heidelberg (1998)
12. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
13. Suzuki, S., Be, K.: Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing 30(1), 32–46 (1985)
14. Hu, M.K.: Visual pattern recognition by moment invariants. IRE Transactions on Information Theory 8(2), 179–187 (1962)
15. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011)
16. Kleesiek, J., Badde, S., Wermter, S., Engel, A.K.: What do Objects Feel Like? Active Perception for a Humanoid Robot. In: Proceedings of the 4th International Conference on Agents and Artificial Intelligence (ICAART), vol. 1, pp. 64–73 (2012)
17. Meltzoff, A.N., Meltzoff, A.N., Moore, M.K.: Explaining facial imitation: a theoretical model. Early Development and Parenting 6, 179–192 (1997)

18. Gibson, J.J.: The theory of affordances. In: Shaw, R., Bransford, J. (eds.) Perceiving, Acting, and Knowing: Toward an Ecological Psychology, pp. 67–82. Erlbaum, Hillsdale (1977)
19. Ogata, T., Ohba, H., Tani, J., Komatani, K., Okuno, H.G.: Extracting multi-modal dynamics of objects using RNNPB. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Edmonton, pp. 160–165 (2005)
20. Körding, K.P., Wolpert, D.M.: Bayesian integration in sensorimotor learning. Nature 427(6971), 244–247 (2004)

Meta-learning of Exploration/Exploitation Strategies: The Multi-armed Bandit Case

Francis Maes, Louis Wehenkel, and Damien Ernst

University of Liège, Dept. of Electrical Engineering and Computer Science
Institut Montefiore, B28, B-4000, Liège, Belgium
{francis.maes,L.Wehenkel,dernst}@ulg.ac.be

Abstract. The exploration/exploitation (E/E) dilemma arises naturally in many subfields of Science. Multi-armed bandit problems formalize this dilemma in its canonical form. Most current research in this field focuses on generic solutions that can be applied to a wide range of problems. However, in practice, it is often the case that a form of prior information is available about the specific class of target problems. Prior knowledge is rarely used in current solutions due to the lack of a systematic approach to incorporate it into the E/E strategy.

To address a specific class of E/E problems, we propose to proceed in three steps: (i) model prior knowledge in the form of a probability distribution over the target class of E/E problems; (ii) choose a large hypothesis space of candidate E/E strategies; and (iii), solve an optimization problem to find a candidate E/E strategy of maximal average performance over a sample of problems drawn from the prior distribution.

We illustrate this meta-learning approach with two different hypothesis spaces: one where E/E strategies are numerically parameterized and another where E/E strategies are represented as small symbolic formulas. We propose appropriate optimization algorithms for both cases. Our experiments, with two-armed "Bernoulli" bandit problems and various playing budgets, show that the meta-learnt E/E strategies outperform generic strategies of the literature (UCB1, UCB1-TUNED, UCB-V, KL-UCB and ϵ_n-GREEDY); they also evaluate the robustness of the learnt E/E strategies, by tests carried out on arms whose rewards follow a truncated Gaussian distribution.

Keywords: Exploration-exploitation dilemma, Prior knowledge, Multi-armed bandit problems, Reinforcement learning.

1 Introduction

Exploration versus exploitation (E/E) dilemmas arise in many sub-fields of Science, and in related fields such as artificial intelligence, finance, medicine and engineering. In its most simple version, the multi-armed bandit problem formalizes this dilemma as follows [1]: a gambler has T coins, and at each step he may choose among one of K slots (or arms) to allocate one of these coins, and then earns some money (his reward) depending on the response of the machine he selected. Each arm response is characterized by an unknown probability distribution that is constant over time. The goal of the gambler is to collect the largest cumulated reward once he has exhausted his coins (i.e.

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 100–115, 2013.

after T plays). A rational (and risk-neutral) gambler knowing the reward distributions of the K arms would play at every stage an arm with maximal expected reward, so as to maximize his expected cumulative reward (irrespectively of the number K of arms, his number T of coins, and the variances of the reward distributions). When reward distributions are unknown, it is less trivial to decide how to play optimally since two contradictory goals compete: *exploration* consists in trying an arm to acquire knowledge on its expected reward, while *exploitation* consists in using the current knowledge to decide which arm to play. How to balance the effort towards these two goals is the essence of the E/E dilemma, which is specially difficult when imposing a finite number of playing opportunities T.

Most theoretical works about multi-armed bandit problem have focused on the design of generic E/E strategies which are provably optimal in asymptotic conditions (large T), while assuming only very unrestrictive conditions on the reward distributions (e.g., bounded support). Among these, some strategies work by computing at every play a quantity called "upper confidence index" for each arm that depends on the rewards collected so far by this arm, and by selecting for the next play (or round of plays) the arm with the highest index. Such E/E strategies are called *index-based policies* and have been initially introduced by [2] where the indices were difficult to compute. More easy to compute indices where proposed later on [3–5].

Index-based policies typically involve hyper-parameters whose values impact their relative performances. Usually, when reporting simulation results, authors manually tuned these values on problems that share similarities with their test problems (e.g., the same type of distributions for generating the rewards) by running trial-and-error simulations [4, 6]. By doing so, they actually used prior information on the problems to select the hyper-parameters.

Starting from these observations, we elaborated an approach for learning in a reproducible way good policies for playing multi-armed bandit problems over finite horizons. This approach explicitly models and then exploits the prior information on the target set of multi-armed bandit problems. We assume that this prior knowledge is represented as a distribution over multi-armed bandit problems, from which we can draw any number of training problems. Given this distribution, meta-learning consists in searching in a chosen set of candidate E/E strategies one that yields optimal expected performances. This approach allows to automatically tune hyper-parameters of existing index-based policies. But, more importantly, it opens the door for searching within much broader classes of E/E strategies one that is optimal for a given set of problems compliant with the prior information. We propose two such hypothesis spaces composed of index-based policies: in the first one, the index function is a linear function of features and whose meta-learnt parameters are real numbers, while in the second one it is a function generated by a grammar of symbolic formulas.

We empirically show, in the case of Bernoulli arms, that when the number K of arms and the playing horizon T are fully specified a priori, learning enables to obtain policies that significantly outperform a wide range of previously proposed generic policies (UCB1, UCB1-TUNED, UCB2, UCB-V, KL-UCB and ϵ_n-GREEDY), even after careful tuning. We also evaluate the robustness of the learned policies with respect to erroneous prior assumptions, by testing the E/E strategies learnt for Bernoulli arms on bandits with rewards following a truncated Gaussian distribution.

The ideas presented in this paper take their roots in two previously published papers. The idea of learning multi-armed bandit policies using global optimization and numerically parameterized index-based policies was first proposed in [7]. Searching good multi-armed bandit policies in a formula space was first proposed in [8]. Compared to this previous work, we adopt here a unifying perspective, which is the learning of E/E strategies from prior knowledge. We also introduce an improved optimization procedure for formula search, based on equivalence classes identification and on a pure exploration multi-armed problem formalization.

This paper is structured as follows. We first formally define the multi-armed bandit problem and introduce index-based policies in Section 2. Section 3 formally states of E/E strategy learning problem. Section 4 and Section 5 present the numerical and symbolic instantiation of our learning approach, respectively. Section 6 reports on experimental results. Finally, we conclude and present future research directions in Section 7.

2 Multi-armed Bandit Problem and Policies

We now formally describe the (discrete) multi-armed bandit problem and the class of index-based policies.

2.1 The Multi-armed Bandit Problem

We denote by $i \in \{1, 2, \ldots, K\}$ the ($K \geq 2$) arms of the bandit problem, by ν_i the reward distribution of arm i, and by μ_i its expected value; b_t is the arm played at round t, and $r_t \sim \nu_{b_t}$ is the obtained reward. $H_t = [b_1, r_1, b_2, r_2, \ldots, b_t, r_t]$ is a vector that gathers the history over the first t plays, and we denote by $\mathcal{H}$ the set of all possible histories of any length t. An E/E strategy (or policy) $\pi : \mathcal{H} \rightarrow \{1, 2, \ldots, K\}$ is an algorithm that processes at play t the vector H_{t-1} to select the arm $b_t \in \{1, 2, \ldots, K\}$: $b_t = \pi(H_{t-1})$.

The regret of the policy π after T plays is defined by: $R_T^\pi = T\mu^* - \sum_{t=1}^{T} r_t$, where $\mu^* = \max_k \mu_k$ refers to the expected reward of the optimal arm. The expected value of the regret represents the expected loss due to the fact that the policy does not always play the best machine. It can be written as:

$$\mathsf{E}\{R_T^\pi\} = \sum_{k=1}^{K} \mathsf{E}\{T_k(T)\}(\mu^* - \mu_k) , \tag{1}$$

where $T_k(T)$ denotes the number of times the policy has drawn arm k on the first T rounds.

The multi-armed bandit problem aims at finding a policy π^* that for a given K minimizes the expected regret (or, in other words, maximizes the expected reward), ideally for any T and any $\{\nu_i\}_{i=1}^{K}$.

Algorithm 1. Generic index-based discrete bandit policy

```
Given scoring function $index : \mathcal{H} \times \{1, 2, \ldots, K\} \to \mathbb{R}$,
for $t = 1$ to $K$ do
    Play bandit $b_t = t$                    ▷ Initialization: play each bandit once
    Observe reward $r_t$
end for
for $t = K$ to $T$ do
    Play bandit $b_t = \mathrm{argmax}_{k \in \{1,2,\ldots,K\}} \, index(H^k_{t-1}, t)$
    Observe reward $r_t$
end for
```

2.2 Index-Based Bandit Policies

Index-based bandit policies are based on a ranking $index$ that computes for each arm k a numerical value based on the sub-history of responses H^k_{t-1} of that arm gathered at time t. These policies are sketched in Algorithm 1 and work as follows. During the first K plays, they play sequentially the machines $1, 2, \ldots, K$ to perform initialization. In all subsequent plays, these policies compute for every machine k the score $index(H^k_{t-1}, t) \in \mathbb{R}$ that depends on the observed sub-history H^k_{t-1} of arm k and possibly on t. At each step t, the arm with the largest score is selected (ties are broken at random).

Here are some examples of popular index functions:

$$index^{\mathrm{UCB1}}(H^k_{t-1}, t) = \overline{r}_k + \sqrt{\frac{C \ln t}{t_k}} \tag{2}$$

$$index^{\mathrm{UCB1\text{-}T{\scriptsize UNED}}}(H^k_{t-1}, t) = \overline{r}_k + \sqrt{\frac{\ln t}{t_k} \min\left(1/4, \overline{\sigma}_k + \sqrt{\frac{2 \ln t}{t_k}}\right)} \tag{3}$$

$$index^{\mathrm{UCB1\text{-}N{\scriptsize ORMAL}}}(H^k_{t-1}, t) = \overline{r}_k + \sqrt{16 \frac{t_k \overline{\sigma}_k^2}{t_k - 1} \frac{\ln(t-1)}{t_k}} \tag{4}$$

$$index^{\mathrm{UCB\text{-}V}}(H^k_{t-1}, t) = \overline{r}_k + \sqrt{\frac{2 \overline{\sigma}_k^2 \zeta \ln t}{t_k}} + c \frac{3 \zeta \ln t}{t_k} \tag{5}$$

where $\overline{r}_k$ and $\overline{\sigma}_k$ are the mean and standard deviation of the rewards so far obtained from arm k and t_k is the number of times it has been played.

Policies UCB1, UCB1-Tuned and UCB1-Normal[1] have been proposed by [4]. UCB1 has one parameter $C > 0$ whose typical value is 2. Policy UCB-V has been proposed by [5] and has two parameters $\zeta > 0$ and $c > 0$. We refer the reader to [4, 5] for detailed explanations of these parameters. Note that these index function are the sum of an exploitation term to give preference on arms with high reward mean ($\overline{r}_k$) and an exploration term that aims at playing arms to gather more information on their underlying reward distribution (which is typically an upper confidence term).

[1] Note that this index-based policy does not strictly fit inside Algorithm 1 as it uses an additional condition to play bandits that were not played since a long time.

3 Learning Exploration/Exploitation Strategies

Instead of relying on a fixed E/E strategy to solve a given class of problems, we propose a systematic approach to exploit prior knowledge by learning E/E strategies in a problem-driven way. We now state our learning approach in abstract terms.

Prior knowledge is represented as a distribution $\mathcal{D}_P$ over bandit problems $P = (\nu_1, \ldots, \nu_K)$. From this distribution, we can sample as many training problems as desired. In order to learn E/E strategies exploiting this knowledge, we rely on a parametric family of candidate strategies $\Pi_\Theta \subset \{1, 2, \ldots, K\}^{\mathcal{H}}$ whose members are policies π_θ that are fully defined given parameters $\theta \in \Theta$. Given Π_Θ, the learning problem aims at solving:

$$\theta^* = \operatorname*{argmin}_{\theta \in \Theta} \mathsf{E}_{P \sim \mathcal{D}_P}\{\mathsf{E}\{R^\pi_{P,T}\}\} , \tag{6}$$

where $\mathsf{E}\{R^\pi_{P,T}\}$ is the expected cumulative regret of π on problem P and where T is the (a-priori given) time playing horizon. Solving this minimization problem is non trivial since it involves an expectation over an infinite number of problems. Furthermore, given a problem P, computing $\mathsf{E}\{R^\pi_{P,T}\}$ relies on the expected values of $T_k(T)$, which we cannot compute exactly in the general case. Therefore, we propose to approximate the expected cumulative regret by the empirical mean regret over a finite set of training problems $P^{(1)}, \ldots, P^{(N)}$ from $\mathcal{D}_P$:

$$\theta^* = \operatorname*{argmin}_{\theta \in \Theta} \Delta(\pi_\theta) \text{ where } \Delta(\pi) = \frac{1}{N} \sum_{i=1}^{N} R^\pi_{P^{(i)},T} , \tag{7}$$

and where $R^{\pi_\theta}_{P^{(i)},T}$ values are estimated performing a single trajectory of π_θ on problem P. Note that the number of training problems N will typically be large in order to make the variance $\Delta(\cdot)$ reasonably small.

In order to instantiate this approach, two components have to be provided: the hypothesis space Π_Θ and the optimization algorithm to solve Eq. 7. The next two sections describe different instantiations of these components.

4 Numeric Parameterization

We now instantiate our meta-learning approach by considering E/E strategies that have numerical parameters.

4.1 Policy Search Space

To define the parametric family of candidate policies Π_Θ, we use index functions expressed as linear combinations of history features. These index functions rely on an *history feature function* $\phi : \mathcal{H} \times \{1, 2, \ldots, K\} \to \mathbb{R}^d$, that describes the history *w.r.t.* a given arm as a vector of scalar features. Given the function $\phi(\cdot, \cdot)$, index functions are defined by

$$index_\theta(H_t, k) = \langle \theta, \phi(H_t, k) \rangle ,$$

where $\theta \in \mathbb{R}^d$ are parameters and $\langle \cdot, \cdot \rangle$ is the classical dot product operator. The set of candidate policies Π_Θ is composed of all index-based policies obtained with such index functions given parameters $\theta \in \mathbb{R}^d$.

History features may describe any aspect of the history, including empirical reward moments, current time step, arm play counts or combinations of these variables. The set of such features should not be too large to avoid parameter estimation difficulties, but it should be large enough to provide the support for a rich set of E/E strategies. We here propose one possibility for defining the history feature function, that can be applied to any multi-armed problem and that is shown to perform well in Section 6.

To compute $\phi(H_t, k)$, we first compute the following four variables: $v_1 = \sqrt{\ln t}$, $v_2 = 1/\sqrt{t_k}$, $v_3 = \overline{r}_k$ and $v_4 = \overline{\sigma}_k$, i.e. the square root of the logarithm of the current time step, the inverse square root of the number of times arm k has been played, the empirical mean and standard deviation of the rewards obtained so far by arm k.

Then, these variables are multiplied in different ways to produce features. The number of these combinations is controlled by a parameter P whose default value is 1. Given P, there is one feature $f_{i,j,k,l}$ per possible combinations of values of $i, j, k, l \in \{0, \ldots, P\}$, which is defined as follows: $f_{i,j,k,l} = v_1^i v_2^j v_3^k v_4^l$.

In other terms, there is one feature per possible polynomial up to degree P using variables $v_1, \ldots, v_4$. In the following, we denote POWER-1 (resp., POWER-2) the policy learned using function $\phi(H_t, k)$ with parameter $P = 1$ (resp., $P = 2$). The index function that underlies these policies can be written as following:

$$index^{power-P}(H_t, k) = \sum_{i=0}^{P} \sum_{j=0}^{P} \sum_{k=0}^{P} \sum_{l=0}^{P} \theta_{i,j,k,l} v_1^i v_2^j v_3^k v_4^l \tag{8}$$

where $\theta_{i,j,k,l}$ are the learned parameters. The POWER-1 policy has 16 such parameters and the POWER-2 has 81 parameters.

4.2 Optimisation Algorithm

We now discuss the optimization of Equation 7 in the case of our numerical parameterization. Note that the objective function we want to optimize, in addition to being stochastic, has a complex relation with the parameters θ. A slight change in the parameter vector θ may lead to significantly different bandit episodes and expected regret values. Local optimization approaches may thus not be appropriate here. Instead, we suggest the use of derivative-free global optimization algorithms.

In this work, we use a powerful, yet simple, class of global optimization algorithms known as *cross-entropy* and also known as *Estimation of Distribution Algorithms* (EDA) [9]. EDAs rely on a probabilistic model to describe promising regions of the search space and to sample good candidate solutions. This is performed by repeating iterations that first *sample* a population of n_p candidates using the *current* probabilistic model and then *fit* a *new* probabilistic model given the $b < n_p$ best candidates.

Any kind of probabilistic model may be used inside an EDA. The simplest form of EDAs uses one marginal distribution per variable to optimize and is known as the *univariate marginal distribution algorithm* [10]. We have adopted this approach by using

Algorithm 2. EDA-based learning of a discrete bandit policy

```
Given the number of iterations $i_{max}$,
Given the population size $n_p$,
Given the number of best elements $b$,
Given a sample of training bandit problems $P^{(1)}, \ldots, P^{(N)}$,
Given an history-features function $\phi(\cdot,\cdot) \in \mathbb{R}^d$,

Set $\mu_p = 0, \sigma_p^2 = 1$ ,$\forall p \in [1,d]$            ▷ Initialize with normal Gaussians
for $i \in [1, i_{max}]$ do
    for $j \in [1, n_p]$ do                                    ▷ Sample and evaluate new population
        for $p \in [1,d]$ do
            $\theta_p \leftarrow$ sample from $\mathcal{N}(\mu_p, \sigma_p^2)$
        end for
        Estimate $\Delta(\pi_\theta)$ and store result $(\theta, \Delta(\pi_\theta))$
    end for
    Select $\{\theta^{(1)}, \ldots, \theta^{(b)}\}$ the $b$ best candidate $\theta$ vectors w.r.t. their $\Delta(\cdot)$ score
    $\mu_p \leftarrow \frac{1}{b}\sum_{j=1}^{b} \theta_p^{(j)}$ ,$\forall p \in [1,d]$            ▷ Learn new Gaussians
    $\sigma_p^2 \leftarrow \frac{1}{b}\sum_{j=1}^{b} (\theta_p^{(j)} - \mu_p)^2$ ,$\forall p \in [1,d]$
end for
return The policy $\pi_\theta$ that led to the lowest observed value of $\Delta(\pi_\theta)$
```

one Gaussian distribution $\mathcal{N}(\mu_p, \sigma_p^2)$ for each parameter θ_p. Although this approach is simple, it proved to be quite effective experimentally to solve Equation 7. The full details of our EDA-based policy learning procedure are given by Algorithm 2. The initial distributions are standard Gaussian distributions $\mathcal{N}(0,1)$. The policy that is returned corresponds to the θ parameters that led to the lowest observed value of $\Delta(\pi_\theta)$.

5 Symbolic Parametrization

The index functions from the literature depend on the current time step t and on three statistics extracted from the sub-history H_{t-1}^k : $\overline{r}_k$, $\overline{\sigma}_k$ and t_k. We now propose a second parameterization of our learning approach, in which we consider all index functions that can be constructed using small formulas built upon these four variables.

5.1 Policy Search Space

We consider index functions that are given in the form of small, closed-form formulas. Closed-form formulas have several advantages: they can be easily computed, they can formally be analyzed and they are easily interpretable.

Let us first explicit the set of formulas $\mathbb{F}$ that we consider in this paper. A formula $F \in \mathbb{F}$ is:

- either a binary expression $F = B(F', F'')$, where B belongs to a set of binary operators $\mathbb{B}$ and F' and F'' are also formulas from $\mathbb{F}$,
- or a unary expression $F = U(F')$ where U belongs to a set of unary operators $\mathbb{U}$ and $F' \in \mathbb{F}$,

$$
\begin{aligned}
F &::= B(F,F) \mid U(F) \mid V \mid C\\
B &::= + \mid - \mid \times \mid \div \mid min \mid max\\
U &::= sqrt \mid ln \mid abs \mid opposite \mid inverse\\
V &::= \overline{r}_k \mid \overline{\sigma}_k \mid t_k \mid t\\
C &::= 1, 2, 3, 5, 7
\end{aligned}
$$

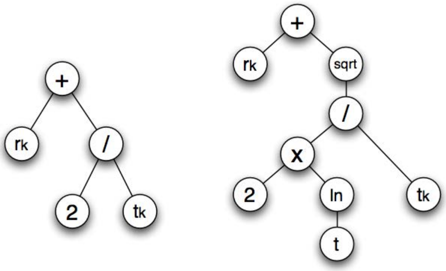


Fig. 1. The grammar used for generating candidate index functions and two example formula parse trees corresponding to $\overline{r}_k + 2/t_k$ and $\overline{r}_k + \sqrt{2ln(t)/t_k}$

- or an atomic variable $F = V$, where V belongs to a set of variables $\mathbb{V}$,
- or a constant $F = C$, where C belongs to a set of constants $\mathbb{C}$.

In the following, we consider a set of operators and constants that provides a good compromise between high expressiveness and low cardinality of $\mathbb{F}$. The set of binary operators considered in this paper $\mathbb{B}$ includes the four elementary mathematic operations and the min and max operators: $\mathbb{B} = \{+, -, \times, \div, \min, \max\}$. The set of unary operators $\mathbb{U}$ contains the square root, the logarithm, the absolute value, the opposite and the inverse: $\mathbb{U} = \left\{\sqrt{.}, \ln(.), |.|, -., \frac{1}{.}\right\}$. The set of variables $\mathbb{V}$ is: $\mathbb{V} = \{\overline{r}_k, \overline{\sigma}_k, t_k, t\}$. The set of constants $\mathbb{C}$ has been chosen to maximize the number of different numbers representable by small formulas. It is defined as $\mathbb{C} = \{1, 2, 3, 5, 7\}$.

Figure 1 summarizes our grammar of formulas and gives two examples of index functions. The length of a formula $length(f)$ is the number of symbols occurring in the formula. For example, the length of $\overline{r}_k + 2/t_k$ is 5 and the length of $\overline{r}_k + \sqrt{2 \times ln(t)/t_k}$ is 9. Let L be a given maximal length. Θ is the subset of formulas whose length is no more than L: $\Theta = \{f | length(f) \leq L\}$ and Π_Θ is the set of index-based policies whose index functions are defined by formulas $f \in \Theta$.

5.2 Optimisation Algorithm

We now discuss the optimization of Equation 7 in the case of our symbolic parameterization. First, notice that several different formulas can lead to the same policy. For example, any increasing function of $\overline{r}_k$ defines the greedy policy, which always selects the arm that is believed to be the best. Examples of such functions in our formula search space include $\overline{r}_k, \overline{r}_k \times 2, \overline{r}_k \times \overline{r}_k$ or $\sqrt{\overline{r}_k}$.

Since it is useless to evaluate equivalent policies multiple times, we propose the following two-step approach. First, the set Θ is partitioned into equivalence classes, two formulas being equivalent if and only if they lead to the same policy. Then, Equation 7 is solved over the set of equivalence classes (which is typically one or two orders of magnitude smaller than the initial set Θ).

Partitioning Θ. This task is far from trivial: given a formula, equivalent formulas can be obtained through commutativity, associativity, operator-specific rules and through any increasing transformation. Performing this step exactly involves advanced static analysis of the formulas, which we believe to be a very difficult solution to implement.

Instead, we propose a simple approximated solution, which consists in discriminating formulas by comparing how they rank (in terms of values returned by the formula) a set of d random samples of the variables $\overline{r}_k, \overline{\sigma}_k, t_k, t$. More formally, the procedure is the following:

1. we first build Θ, the space of all formulas f such that $length(f) \leq L$;
2. for $i = 1 \dots d$, we uniformly draw (within their respective domains) some random realizations of the variables $\overline{r}_k, \overline{\sigma}_k, t_k, t$ that we concatenate into a vector Θ_i;
3. we cluster all formulas from Θ according to the following rule: two formulas F and F' belong to the same cluster if and only if they rank all the Θ_i points in the same order, i.e.: $\forall i, j \in \{1, \dots, d\}, i \neq j, F(\Theta_i) \geq F(\Theta_j) \iff F'(\Theta_i) \geq F'(\Theta_j)$. Formulas leading to invalid index functions (caused for instance by division by zero or logarithm of negative values) are discarded;
4. among each cluster, we select one formula of minimal length;
5. we gather all the selected minimal length formulas into an approximated reduced set of formulas $\tilde{\Theta}$.

In the following, we denote by M the cardinality of the approximate set of formulas $\tilde{\Theta} = \{f_1, \dots, f_M\}$.

Optimization Algorithm. A naive approach for finding the best formula $f^* \in \tilde{\Theta}$ would be to evaluate $\Delta(f)$ for each formula $f \in \tilde{\Theta}$ and simply return the best one. While extremely simple to implement, such an approach could reveal itself to be time-inefficient in case of spaces $\tilde{\Theta}$ of large cardinality.

Preliminary experiments have shown us that $\tilde{\Theta}$ contains a majority of formulas leading to relatively bad performing index-based policies. It turns out that relatively few samples of $R^{\pi}_{P^{(i)},T}$ are sufficient to reject with high confidence these badly performing formulas. In order to exploit this idea, a natural idea is to formalize the search for the best formula as another multi-armed bandit problem. To each formula $F_k \in \tilde{\Theta}$, we associate an arm. Pulling the arm k consists in selecting a training problem $P^{(i)}$ and in running one episode with the index-based policy whose index formula is f_k. This leads to a reward associated to arm k whose value is the quantity $-R^{\pi}_{P^{(i)},T}$ observed during the episode. The purpose of multi-armed bandit algorithms is here to process the sequence of observed rewards to select in a smart way the next formula to be tried so that when the budget of pulls has been exhausted, one (or several) high-quality formula(s) can be identified.

In the formalization of Equation 7 as a multi-armed bandit problem, only the quality of the finally suggested arm matters. How to select arms so as to identify the best one in a finite amount of time is known as the *pure exploration* multi-armed bandit problem [11]. It has been shown that index-based policies based on upper confidence bounds were good policies for solving pure exploration bandit problems. Our optimization procedure works as follows: we use a bandit algorithm such as UCB1-TUNED during a given number of steps and then return the policy that corresponds to the formula f_k with highest expected reward $\overline{r}_k$. The problem instances are selected depending on the number of times the arm has been played so far: at each step, we select the training problem $P^{(i)}$ with $i = 1 + (t_k \bmod N)$.

In our experiments, we estimate that our multi-armed bandit approach is one hundred to one thousand times faster than the naive Monte Carlo optimization procedure, which clearly demonstrates the benefits of this approach. Note that this idea could also be relevant to our numerical case. The main difference is that the corresponding multi-armed bandit problem relies on a continuous-arm space. Although some algorithms have already been proposed to solve such multi-armed bandit problems [12], how to scale these techniques to problems with hundreds or thousands parameters is still an open research question. Progresses in this field could directly benefit our numerical learning approach.

6 Numerical Experiments

We now illustrate the two instances of our learning approach by comparing learned policies against a number of generic previously proposed policies in a setting where prior knowledge is available about the target problems. We show that in both cases, learning enables to obtain exploration/exploitation strategies significantly outperforming all tested generic policies.

6.1 Experimental Protocol

We compare learned policies against generic policies. We distinguish between *untuned generic policies* and *tuned generic policies*. The former are either policies that are parameter-free or policies used with default parameters suggested in the literature, while the latter are generic policies whose hyper-parameters were tuned using Algorithm 2.

Training and Testing. To illustrate our approach, we consider the scenario where the number of arms K, the playing horizon T and the kind of distributions ν_k are known a priori and where the parameters of these distributions are missing information. Since we are learning policies, care should be taken with generalization issues. As usual in supervised machine learning, we use a training set which is distinct from the testing set. The training set is composed of $N = 100$ bandit problems sampled from a given distribution over bandit problems $\mathcal{D}_P$ whereas the testing set contains another 10000 problems drawn from this distribution. To study the robustness of our policies w.r.t. wrong prior information, we also report their performance on a set of 10000 problems drawn from another distribution $\mathcal{D}'_P$ with different kinds of distributions ν_k. When computing $\Delta(\pi_\theta)$, we estimate the regret for each of these problems by averaging results overs 100 runs. One calculation of $\Delta(\pi_\theta)$ thus involves simulating 10^4 (resp. 10^6) bandit episodes during training (resp. testing).

Problem Distributions. The distribution $\mathcal{D}_P$ is composed of two-armed bandit problems with Bernoulli distributions whose expectations are uniformly drawn from $[0, 1]$. Hence, in order to sample a bandit problem from $\mathcal{D}_P$, we draw the expectations p_1 and p_2 uniformly from $[0, 1]$ and return the bandit problem with two Bernoulli arms that have expectations p_1 and p_2, respectively. In the second distribution $\mathcal{D}'_P$, the reward distributions ν_k are changed by Gaussian distributions truncated to the interval $[0, 1]$. In order to sample one problem from $\mathcal{D}'_P$, we select a mean and a standard deviation for each arm uniformly in range $[0, 1]$. Rewards are then sampled using a rejection sampling approach: samples are drawn from the corresponding Gaussian distribution until obtaining a value that belongs to the interval $[0, 1]$.

Generic Policies. We consider the following generic policies: the ϵ_n-GREEDY policy as described in [4], the policies introduced by [4]: UCB1, UCB1-TUNED, UCB1-NORMAL and UCB2, the policy KL-UCB introduced in [13] and the policy UCB-V proposed by [5]. Except ϵ_n-GREEDY, all these policies belong to the family of index-based policies discussed previously. UCB1-TUNED and UCB1-NORMAL are parameter-free policies designed for bandit problems with Bernoulli distributions and for problems with Gaussian distributions respectively. All the other policies have hyper-parameters that can be tuned to improve the quality of the policy. ϵ_n-GREEDY has two parameters $c > 0$ and $0 < d < 1$, UCB2 has one parameter $0 < \alpha < 1$, KL-UCB has one parameter $c \geq 0$ and UCB-V has two parameters $\zeta > 0$ and $c > 0$. We refer the reader to [4, 5, 13] for detailed explanations of these parameters.

Learning Numerical Policies. We learn policies using the two parameterizations POWER-1 and POWER-2 described in Section 4.1. Note that tuning generic policies is a particular case of learning with numerical parameters and that both learned policies and tuned generic policies make use of the same prior knowledge. To make our comparison between these two kinds of policies fair, we always use the same training procedure, which is Algorithm 2 with $i_{max} = 100$ iterations, $n_p = max(8d, 40)$ candidate policies per iteration and $b = n_p/4$ best elements, where d is the number of parameters to optimize. Having a linear dependency between n_p and d is a classical choice when using EDAs [14]. Note that, in most cases the optimization is solved in a few or a few tens iterations. Our simulations have shown that $i_{max} = 100$ is a careful choice for ensuring that the optimization has enough time to properly converge. For the baseline policies where some default values are advocated, we use these values as initial expectation of the EDA Gaussians. Otherwise, the initial Gaussians are centered on zero. Nothing is done to enforce the EDA to respect the constraints on the parameters (e.g., $c > 0$ and $0 < d < 1$ for ϵ_n-GREEDY). In practice, the EDA automatically identifies interesting regions of the search space that respect these constraints.

Learning Symbolic Policies. We apply our symbolic learning approach with a maximal formula length of $L = 7$, which leads to a set of $|\Theta| \approx 33,5$ millions of formulas. We have applied the approximate partitioning approach described in Section 5.2 on these formulas using $d = 1024$ samples to discriminate among strategies. This has resulted in $\approx 9,5$ million invalid formulas and $M = 99020$ distinct candidate E/E strategies (i.e. distinct formula equivalence classes). To identify the best of those distinct strategies, we apply the UCB1-TUNED algorithm for 10^7 steps. In our experiments, we report the two best found policies, which we denote FORMULA-1 and FORMULA-2.

6.2 Performance Comparison

Table 1 reports the results we obtain for untuned generic policies, tuned generic policies and learned policies on distributions $\mathcal{D}_P$ and $\mathcal{D}'_P$ with horizons $T \in \{10, 100, 1000\}$. For both tuned and learned policies, we consider three different training horizons $\{10, 100, 1000\}$ to see the effect of a mismatch between the training and the testing horizon.

Table 1. Mean expected regret of untuned, tuned and learned policies on Bernoulli and Gaussian bandit problems. Best scores in each of these categories are shown in bold. Scores corresponding to policies that are tested on the same horizon T than the horizon used for training/tuning are shown in italics.

Policy	Training Horizon	Parameters	Bernoulli			Gaussian		
			T=10	T=100	T=1000	T=10	T=100	T=1000
		Untuned generic policies						
UCB1	-	$C = 2$	1.07	5.57	20.1	1.37	10.6	66.7
UCB1-Tuned	-		**0.75**	**2.28**	**5.43**	**1.09**	6.62	**37.0**
UCB1-Normal	-		1.71	13.1	31.7	1.65	13.4	58.8
UCB2	-	$\alpha = 10^{-3}$	0.97	3.13	7.26	1.28	7.90	40.1
UCB-V	-	$c = 1, \zeta = 1$	1.45	8.59	25.5	1.55	12.3	63.4
KL-UCB	-	$c = 0$	0.76	2.47	6.61	1.14	7.66	43.8
KL-UCB	-	$c = 3$	0.82	3.29	9.81	1.21	8.90	53.0
ϵ_n-Greedy	-	$c = 1, d = 1$	1.07	3.21	11.5	1.20	**6.24**	41.4
		Tuned generic policies						
UCB1	T=10	$C = 0.170$	*0.74*	**2.05**	4.85	1.05	6.05	32.1
	T=100	$C = 0.173$	0.74	***2.05***	4.84	1.05	6.06	32.3
	T=1000	$C = 0.187$	0.74	2.08	***4.91***	1.05	6.17	33.0
UCB2	T=10	$\alpha = 0.0316$	*0.97*	3.15	7.39	1.28	7.91	40.5
	T=100	$\alpha = 0.000749$	0.97	*3.12*	7.26	1.33	8.14	40.4
	T=1000	$\alpha = 0.00398$	0.97	3.13	*7.25*	1.28	7.89	40.0
UCB-V	T=10	$c = 1.542, \zeta = 0.0631$	*0.75*	2.36	5.15	**1.01**	5.75	26.8
	T=100	$c = 1.681, \zeta = 0.0347$	0.75	*2.28*	7.07	**1.01**	**5.30**	**27.4**
	T=1000	$c = 1.304, \zeta = 0.0852$	0.77	2.43	*5.14*	1.13	5.99	27.5
KL-UCB	T=10	$c = -1.21$	***0.73***	2.14	5.28	1.12	7.00	38.9
	T=100	$c = -1.82$	**0.73**	*2.10*	5.12	1.09	6.48	36.1
	T=1000	$c = -1.84$	**0.73**	2.10	*5.12*	1.08	6.34	35.4
ϵ_n-Greedy	T=10	$c = 0.0499, d = 1.505$	*0.79*	3.86	32.5	**1.01**	7.31	67.6
	T=100	$c = 1.096, d = 1.349$	0.95	*3.19*	14.8	1.12	6.38	46.6
	T=1000	$c = 0.845, d = 0.738$	1.23	3.48	*9.93*	1.32	6.28	37.7
		Learned numerical policies						
Power-1	T=10	...	***0.72***	2.29	14.0	**0.97**	5.94	49.7
	T=100	*(16 parameters)*	0.77	*1.84*	5.64	1.04	5.13	27.7
	T=1000	...	0.88	2.09	*4.04*	1.17	5.95	28.2
Power-2	T=10	...	***0.72***	2.37	15.7	**0.97**	6.16	55.5
	T=100	*(81 parameters)*	0.76	***1.82***	5.81	1.05	**5.03**	29.6
	T=1000	...	0.83	2.07	***3.95***	1.12	5.61	**27.3**
		Learned symbolic policies						
Formula-1	T=10	$\sqrt{t_k}(\overline{r}_k - 1/2)$	***0.72***	2.37	14.7	**0.96**	5.14	30.4
	T=100	$\overline{r}_k + 1/(t_k + 1/2)$	0.76	***1.85***	8.46	1.12	**5.07**	29.8
	T=1000	$\overline{r}_k + 3/(t_k + 2)$	0.80	2.31	***4.16***	1.23	6.49	26.4
Formula-2	T=10	$\lvert\overline{r}_k - 1/(t_k + t)\rvert$	*0.72*	2.88	22.8	1.02	7.15	66.2
	T=100	$\overline{r}_k + min(1/t_k, log(2))$	0.78	*1.92*	6.83	1.17	5.22	29.1
	T=1000	$1/t_k - 1/(\overline{r}_k - 2)$	1.10	2.62	*4.29*	1.38	6.29	**26.1**

Table 2. Percentage of wins against UCB1-Tuned of generic and learned policies. Best scores are shown in bold.

Policy	T = 10	T = 100	T = 1000	Policy	T = 10	T = 100	T = 1000
Generic policies				*Learned policies*			
UCB1	48.1 %	78.1 %	83.1 %	Power-1	54.6 %	82.3 %	**91.3 %**
UCB2	12.7 %	6.8 %	6.8 %	Power-2	54.2 %	**84.6 %**	90.3 %
UCB-V	38.3 %	57.2 %	49.6 %	Formula-1	**61.7** %	76.8 %	88.1 %
KL-UCB	50.5 %	65.0 %	67.0 %	Formula-2	61.0 %	80.0 %	73.1 %
ϵ_n-Greedy	37.5 %	14.1 %	10.7 %				

Generic Policies. As already pointed out in [4], it can be seen that UCB1-Tuned is particularly well fitted to bandit problems with Bernoulli distributions. It also proves effective on bandit problems with Gaussian distributions, making it nearly always outperform the other untuned policies. By tuning UCB1, we outperform the UCB1-Tuned policy (e.g. 4.91 instead of 5.43 on Bernoulli problems with $T = 1000$). This also sometimes happens with UCB-V. However, though we used a careful tuning procedure, UCB2 and ϵ_n-Greedy do never outperform UCB1-Tuned.

Learned Policies. We observe that when the training horizon is the same as the testing horizon T, the learned policies (Power-1, Power-2, Formula-1 and Formula-2) systematically outperform all generic policies. The overall best results are obtained with Power-2 policies. Note that, due to their numerical nature and due to the large number of parameters, these policies are extremely hard to interpret and to understand. The results related to symbolic policies show that there exist very simple policies that perform nearly as well as these black-box policies. This clearly shows the benefits of our two hypothesis spaces: numerical policies enable to reach very high performances while symbolic policies provide interpretable strategies whose behavior can be more easily analyzed. This interpretability/performance tradeoff is common in machine learning and has been identified several decades ago in the field of supervised learning. It is worth mentioning that, among the 99020 formula equivalence classes, a surprisingly large number of strategies outperforming generic policies were found: when $T = 100$ (resp. $T = 1000$), we obtain about 50 (resp. 80) different symbolic policies outperforming the generic policies.

Robustness w.r.t. the Horizon T. As expected, the learned policies give their best performance when the training and the testing horizons are equal. Policies learned with large training horizon prove to work well also on smaller horizons. However, when the testing horizon is larger than the training horizon, the quality of the policy may quickly degrade (e.g. when evaluating Power-1 trained with $T = 10$ on an horizon $T = 1000$).

Robustness w.r.t. the Kind of Distribution. Although truncated Gaussian distributions are significantly different from Bernoulli distributions, the learned policies most of the time generalize well to this new setting and still outperform all the other generic policies.

A Word on the Learned Symbolic Policies. It is worth noticing that the best index-based policies (Formula-1) found for the two largest horizons ($T = 100$ and $T = 1000$) work in a similar way as the UCB-type policies reported earlier in the literature. Indeed, they also associate to an arm k an index which is the sum of $\overline{r}_k$ and of a positive (optimistic) term that decreases with t_k. However, for the shortest time horizon ($T = 10$), the policy found ($\sqrt{t_k}(\overline{r}_k - \frac{1}{2})$) is totally different from UCB-type policies. With such a policy, only the arms whose empirical reward mean is higher than a given threshold (0.5) have positive index scores and are candidate for selection, i.e. making the scores negative has the effect to kill bad arms. If the $\overline{r}_k$ of an arm is above the threshold, then the index associated with this arm will increase with the number of times it is played and not decrease as it is the case for UCB policies. If all empirical means $\overline{r}_k$ are below the threshold, then for equal reward means, arms that have been less played are preferred. This finding is amazing since it suggests that this optimistic paradigm for multi-armed bandits upon which UCB policies are based may in fact not be adapted at all to a context where the horizon is small.

Percentage of Wins Against UCB1-TUNED. Table 2 gives for each policy, its percentage of wins against UCB1-TUNED, when trained with the same horizon as the test horizon. To compute this percentage of wins, we evaluate the expected regret on each of the 10000 testing problems and count the number of problems for which the tested policy outperforms UCB1-TUNED. We observe that by minimizing the expected regret, our learned policies also reach high values of percentage of wins: 84.6 % for $T = 100$ and 91.3 % for $T = 1000$. Note that, in our approach, it is easy to change the objective function. So if the real applicative aim was to maximize the percentage of wins against UCB1-TUNED, this criterion could have been used directly in the policy optimization stage to reach even better scores.

6.3 Computational Time

We used a C++ based implementation to perform our experiments. In the numerical case with 10 cores at $1.9Ghz$, performing the whole learning of POWER-1 took one hour for $T = 100$ and ten hours for $T = 1000$. In the symbolic case using a single core at $1.9Ghz$, performing the whole learning took 22 minutes for $T = 100$ and a bit less than three hours for $T = 1000$. Note that the fact that symbolic learning is much faster can be explained by two reasons. First, we tuned the EDA algorithm in a very careful way to be sure to find a high quality solution; what we observe is that by using only 10% of this learning time, we already obtain close-to-optimal strategies. The second factor is that our symbolic learning algorithm saves a lot of CPU time by being able to rapidly reject bad strategies thanks to the multi-armed bandit formulation upon which it relies.

7 Conclusions

The approach proposed in this paper for exploiting prior knowledge for learning exploration/exploitation policies has been tested for two-armed bandit problems with Bernoulli reward distributions and when knowing the time horizon. The learned policies were found to significantly outperform other policies previously published in the literature such as UCB1, UCB2, UCB-V, KL-UCB and ϵ_n-GREEDY. The robustness of the learned policies with respect to wrong information was also highlighted, by evaluating them on two-armed bandits with truncated Gaussian reward distribution.

There are in our opinion several research directions that could be investigated for still improving the algorithm for learning policies proposed in this paper. For example, we found out that problems similar to the problem of overfitting met in supervised learning could occur when considering a too large set of candidate polices. This naturally calls for studying whether our learning approach could be combined with regularization techniques. Along this idea, more sophisticated optimizers could also be thought of for identifying in the set of candidate policies, the one which is predicted to behave at best.

The UCB1, UCB2, UCB-V, KL-UCB and ϵ_n-GREEDY policies used for comparison were shown (under certain conditions) to have interesting bounds on their expected regret in asymptotic conditions (very large T) while we did not aim at providing such bounds for our learned policies. It would certainly be relevant to investigate whether similar bounds could be derived for our learned policies or, alternatively, to see how the approach could be adapted so as to target policies offering such theoretical performance guarantees in asymptotic conditions. For example, better bounds on the expected regret could perhaps be obtained by identifying in a set of candidate policies the one that gives the smallest maximal value of the expected regret over this set rather than the one that gives the best average performances.

Finally, while our paper has provided simulation results in the context of the most simple multi-armed bandit setting, our exploration/exploitation policy meta-learning scheme can also in principle be applied to any other exploration-exploitation problem. In this line of research, the extension of this investigation to (finite) Markov Decision Processes studied in [15], suggests already that our approach to meta-learning E/E strategies can be successful on much more complex settings.

References

1. Robbins, H.: Some aspects of the sequential design of experiments. Bulletin of The American Mathematical Society 58, 527–536 (1952)
2. Lai, T., Robbins, H.: Asymptotically efficient adaptive allocation rules. Advances in Applied Mathematics 6, 4–22 (1985)
3. Agrawal, R.: Sample mean based index policies with o(log n) regret for the multi-armed bandit problem. Advances in Applied Mathematics 27, 1054–1078 (1995)
4. Auer, P., Fischer, P., Cesa-Bianchi, N.: Finite-time analysis of the multi-armed bandit problem. Machine Learning 47, 235–256 (2002)
5. Audibert, J.-Y., Munos, R., Szepesvári, C.: Tuning Bandit Algorithms in Stochastic Environments. In: Hutter, M., Servedio, R.A., Takimoto, E. (eds.) ALT 2007. LNCS (LNAI), vol. 4754, pp. 150–165. Springer, Heidelberg (2007)
6. Audibert, J., Munos, R., Szepesvari, C.: Exploration-exploitation trade-off using variance estimates in multi-armed bandits. In: Theoretical Computer Science (2008)
7. Maes, F., Wehenkel, L., Ernst, D.: Learning to play K-armed bandit problems. In: Proc. of the 4th International Conference on Agents and Artificial Intelligence (2012)
8. Maes, F., Wehenkel, L., Ernst, D.: Automatic Discovery of Ranking Formulas for Playing with Multi-armed Bandits. In: Sanner, S., Hutter, M. (eds.) EWRL 2011. LNCS, vol. 7188, pp. 5–17. Springer, Heidelberg (2012)
9. Gonzalez, C., Lozano, J., Larrañaga, P.: Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation. Kluwer Academic Publishers (2002)

10. Pelikan, M., Mühlenbein, H.: Marginal distributions in evolutionary algorithms. In: Proceedings of the 4th International Conference on Genetic Algorithms (1998)
11. Bubeck, S., Munos, R., Stoltz, G.: Pure Exploration in Multi-armed Bandits Problems. In: Gavaldà, R., Lugosi, G., Zeugmann, T., Zilles, S. (eds.) ALT 2009. LNCS, vol. 5809, pp. 23–37. Springer, Heidelberg (2009)
12. Bubeck, S., Munos, R., Stoltz, G., Szepesvári, C.: X-armed bandits. Journal of Machine Learning Research 12, 1655–1695 (2011)
13. Garivier, A., Cappé, O.: The KL-UCB algorithm for bounded stochastic bandits and beyond. CoRR abs/1102.2490 (2011)
14. Rubenstein, R., Kroese, D.: The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simluation, and machine learning. Springer, New York (2004)
15. Castronovo, M., Maes, F., Fonteneau, R., Ernst, D.: Learning exploration/exploitation strategies for single trajectory reinforcement learning. In: Proc. of 10th European Workshop on Reinforcement Learning (2012)

MobEx: A System for Exploratory Search on the Mobile Web

Günter Neumann and Sven Schmeier

DFKI - German Research Center for Artificial Intelligence,
Stuhlsatzenhausweg 3, 66119 Saarbrücken, Germany
{neumann,sven.schmeier}@dfki.de
http://www.dfki.de

Abstract. We present *MobEx*, a mobile touchable application for exploratory search on the mobile web. The system has been implemented for operation on a tablet computer, i.e. an Apple iPad, and on a mobile device, i.e. Apple iPhone or iPod touch. Starting from a topic issued by the user the system collects web snippets that have been determined by a standard search engine in a first step and extracts associated topics to the initial query in an unsupervised way on-demand and highly performant. This process is recursive in priciple as it furthermore determines other topics associated to the newly found ones and so forth. As a result *MobEx* creates a dense web of associated topics that is presented to the user as an interactive topic graph. We consider the extraction of topics as a specific empirical collocation extraction task where collocations are extracted between chunks combined with the cluster descriptions of an online clustering algorithm. Our measure of association strength is based on the pointwise mutual information between chunk pairs which explicitly takes their distance into account. These syntactically–oriented chunk pairs are then semantically ranked and filtered using the cluster descriptions created by a Singular Value Decomposition (SVD) approach. An initial user evaluation shows that this system is especially helpful for finding new interesting information on topics about which the user has only a vague idea or even no idea at all.

Keywords: Web mining, Information extraction, Topic graph exploration, Mobile device.

1 Introduction

Searching the web using standard search engines is still dominated by a passive one–tracked human-computer interaction: a user enters one or more keywords that represent the information of interest and receives a ranked list of documents. However, if the user only has a vague idea of the information in question or just wants to explore the information space, the current search engine paradigm does not provide enough assistance for these kind of searches. The user has to read through the documents and then eventually reformulate the query in order to find new information. This can be a tedious task especially on mobile devices.

In order to overcome this restricted document perspective, and to provide a mobile device searches to "find out about something", we want to help users with the web content exploration process in several ways:

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 116–130, 2013.

1. We consider a user query as a specification of a topic that the user wants to know and learn more about. Hence, the search result is basically a graphical structure of that topic and associated topics that are found.
2. The user can interactively explore this topic graph using a simple and intuitive user interface in order to either learn more about the content of a topic or to interactively expand a topic with newly computed related topics.
3. Nowadays, the mobile web and mobile touchable devices, like smartphones and tablet computers, are getting more and more prominent and widespread. Thus the user might expect a device-adaptable touchable handy human–computer interaction.

In this paper, we present an approach of exploratory web search, that tackles the above mentioned requirements in the following way.

In a first step, the topic graph is computed on the fly from a set of web snippets that has been collected by a standard search engine using the initial user query. Rather than considering each snippet in isolation, all snippets are collected into one document from which the topic graph is computed. We consider each topic as an entity, and the edges are considered as a kind of (hidden) relationship between the connected topics. The content of a topic are the set of snippets it has been extracted from, and the documents retrievable via the snippets' web links.

The topic graph is then displayed either on a tablet computer (in our case an iPad) as touch–sensitive graph or displayed as a stack of touchable text on a smartphone (in our case an iPhone or an iPod touch). By just selecting a node or a text box, the user can either inspect the content of a topic (i.e, the snippets or web pages) or activate the expansion of the topic graph through an on the fly computation of new related topics for the selected node. The user can request information from new topics on basis of previously extracted information by selecting a node from a newly extracted topic graph.

In such a dynamic open–domain information extraction situation, the user expects real–time performance from the underlying technology. The requested information cannot simply be pre–computed, but rather has to be determined in an unsupervised and on–demand manner relative to the current user request. This is why we assume that the relevant information can be extracted from a search engine's *web snippets* directly, and that we can avoid the costly retrieval and processing time for huge amounts of documents. Of course, direct processing of web snippets also poses certain challenges for the Natural Language Processing (NLP) components. Web snippets are usually small text summaries which are automatically created from parts of the source documents and are often only in part linguistically well–formed, cf. [9]. Thus the NLP components are required to possess a high degree of robustness and run–time behavior to process the web snippets in real–time. Since our approach should also be able to process web snippets from different languages (our current application runs for English and German), the NLP components should be easily adaptable to many languages. Finally, no restrictions to the domain of the topic should be pre–supposed, i.e., the system should be able to accept topic queries from arbitrary domains. In order to fulfill all these requirements, we are favoring and exploring the use of shallow and highly data–oriented NLP components. Note that this is not a trivial or obvious design decision, since most of the current prominent information extraction methods advocate deeper NLP components for concept and relation extraction, e.g., syntactic and semantic dependency analysis of complete sentences and the integration of rich linguistic knowledge bases like Word Net.

The paper is organized as follows. In the section 2 we briefly summarize the topic graph extraction process.[1] For the sake of completeness and readability, we present in section 3 details and examples of the user interfaces for the iPad and iPhone, respectively.

A major obstacle of the topic graph extraction process described in section 2 is its purely syntactic nature. Consequently, in section 4, we introduce a semantic clustering approach that helps to improve the quality of the extracted topics. The next sections then describe details of the evaluation of the improved topic extraction process (section 5), and present our current user experience for the iPad and iPhone user interfaces (section 6). Related work is discussed in section 7, before we conclude the paper in section 8.

2 Topic–Driven Exploration of Web Content

The core idea is to compute a set of chunk–pair–distance elements for the N–first web snippets returned by a search engine for the topic Q, and to compute the topic graph from these elements.[2] In general for two chunks, a single chunk–pair–distance element stores the distance between the chunks by counting the number of chunks in-between them. We distinguish elements which have the same words in the same order, but have different distances. For example, (Justin, Selina, 5) is different from (Justin, Selina, 2) and (Selina, Justin, 7).

Initially, a document is created from selected web snippets so that each line contains a complete snippet. Each of these lines is then tagged with Part–of–Speech using the SVMTagger [8] and chunked in the next step.

The chunker recognizes two types of word chains: noun chunks and verb chunks. Each recognized word chain consists of the longest matching sequences of words with the same PoS class, namely noun chains or verb chains, where an element of a noun chain belongs to one of the predefined extended noun tags. Elements of a verb chain only contain verb tags. For English, “word/PoS” expressions that match the regular expression “/(N(N|P))|/VB(N|G)|/IN|/DT” are considered as extended noun tag and for German the expression
“/(N(N|E))|/VVPP|/AP|/ART”. The English Verbs are those whose PoS tag start with VB (and VV in case of German). We are using the tag sets from the Penn treebank (English) and the Negra treebank (German).

The chunk–pair–distance model is computed from the list of noun group chunks.[3] This is fulfilled by traversing the chunks from left to right. For each chunk c_i, a set is computed by considering all remaining chunks and their distance to c_i, i.e.,

[1] This part of the work has partially been presented in [12] and hence will be described and illustrated compactly.

[2] We are using Bing (`http://www.bing.com/`) for web search with N set to max. 1000.

[3] The main purpose of recognizing verb chunks is to improve proper recognition of noun groups. They are ignored when building the topic graph, but see sec. 8.

$(c_i, c_{i+1}, dist_{i(i+1)})$, $(c_i, c_{i+2}, dist_{i(i+2)})$, etc. This is to be done for each chunk list computed for each web snippet. The distance $dist_{ij}$ of two chunks c_i and c_j is computed directly from the chunk list, i.e. we do not count the position of ignored words lying between two chunks.

Finally, we compute the chunk–pair–distance model CPD_M using the frequencies of each chunk, each chunk pair, and each chunk pair distance. CPD_M is used for constructing the topic graph in the final step. Formally, a topic graph $TG = (V, E, A)$ consists of a set V of nodes, a set E of edges, and a set A of node actions. Each node $v \in V$ represents a chunk and is labeled with the corresponding PoS–tagged word group. Node actions are used to trigger additional processing, e.g. displaying the snippets, expanding the graph etc.

The nodes and edges are computed from the chunk–pair–distance elements. Since the number of these elements is quite large (up to several thousands), the elements are ranked according to a weighting scheme which takes into account the frequency information of the chunks and their collocations. More precisely, the weight of a chunk–pair–distance element $cpd = (c_i, c_j, D_{ij})$, with $D_{ij} = \{(freq_1, dist_1), (freq_2, dist_2), ..., (freq_n, dist_n)\}$, is computed based on point–wise mutual information (PMI, cf. [15]) as follows:

$$\begin{aligned} PMI(cpd) &= log_2((p(c_i, c_j)/(p(c_i) * p(c_j))) \\ &= log_2(p(c_i, c_j)) - log_2(p(c_i) * p(c_j)) \end{aligned}$$

where relative frequency is used for approximating the probabilities $p(c_i)$ and $p(c_j)$. For $log_2(p(c_i, c_j))$ we took the (unsigned) polynomials of the corresponding Taylor series using $(freq_k, dist_k)$ in the k-th Taylor polynomial and adding them up:

$$PMI(cpd) = (\sum_{k=1}^{n} \frac{(x_k)^k}{k}) - log_2(p(c_i) * p(c_j))$$

$$, where\ x_k = \frac{freq_k}{\sum_{k=1}^{n} freq_k}$$

The visualized part of the topic graph is then computed from a subset of CPD_M using the m highest ranked chunk–pair–distance elements for fixed c_i. In other words, we restrict the complexity of a topic graph by restricting the number of edges connected to a node.

3 Touchable Interface for Mobile Devices

Today, it is a standard approach to optimize the presentation of a web page, depending on the device it is displayed on, e.g., a standard or mobile web browser. Obviously, the same should hold for graphical user interfaces, and in our case, for the user interfaces designed for iPad and iPhone.

More concretely, the usage of a different mode of presentation and interaction with a topic graph depending on the device at hand, is motivated for the following reasons: For a smartphone the capabilities for displaying touchable text and graphics on one screen are limited mainly due to its relatively small screen size. Our concept for presenting

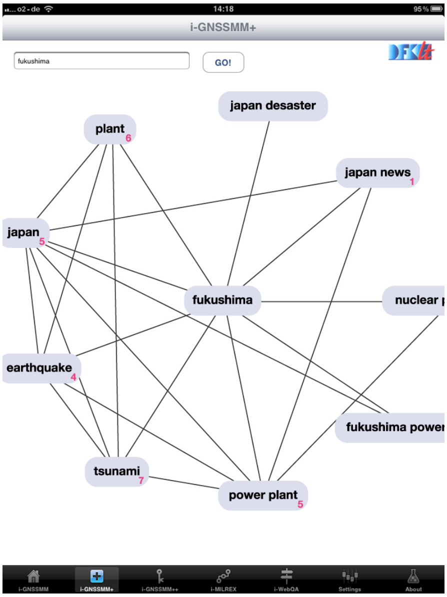


Fig. 1. The topic graph computed from the snippets for the query "Fukushima"

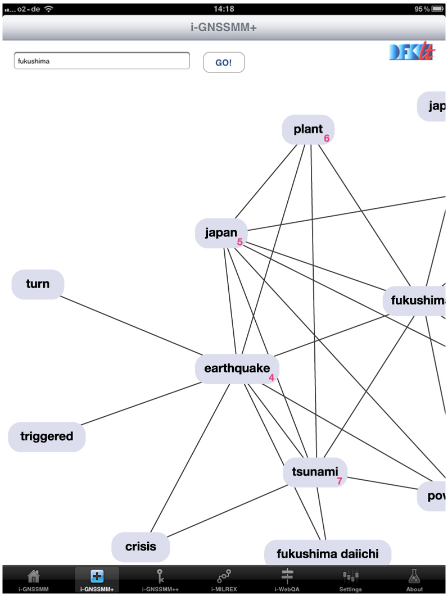


Fig. 2. The topic graph from Fig. 1 has been expanded by a single touch on the node labeled "earthquake"

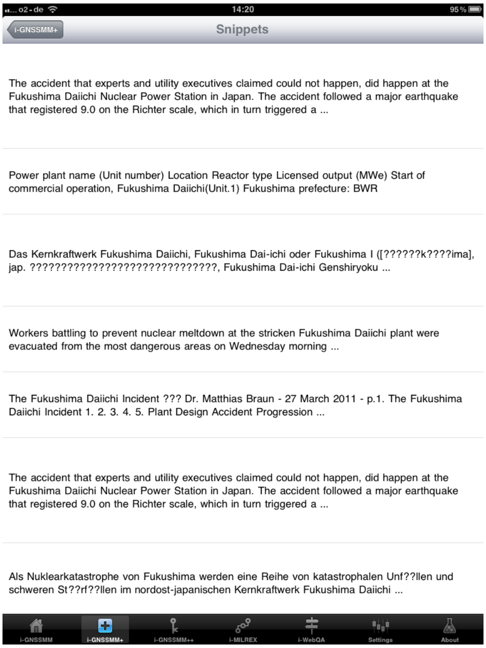
the results consists of a touchable navigation based user interface which allows us to interact easily by single touch and swiping gestures. For a tablet computer with larger screens the intelligent mix of graphics and text makes a software system most appealing to the user. Hence the result presentation consists of a touchable topic graph offering multitouch capabilities like zooming and swiping.

We demonstrate our current solution by the following screenshots which show some results of the search query "Fukushima" running with the current iPad and iPhone user interfaces. In section 6 we present and discuss the outcomes of some user experiments.

3.1 Graph–Based User Interface on the iPad

The screenshot in Fig. 1 shows the topic graph computed from the snippets for the query "Fukushima". The user can double touch on a node to display the associated snippets and web pages. Since a topic graph can be very large, not all nodes are displayed (using the technology described in the previous section). Nodes which can be expanded are marked by the number of hidden immediate nodes. A single touch on such a node expands it, as shown in Fig. 2. A single touch on a node which cannot be expanded automatically adds its label to the initial user query and triggers a new search with that expanded query.

Fig. 2 demonstrates how the topic graph from Fig. 1 has been expanded by a single touch on the node labeled "earthquake". Double touching on the node "fukushima dailchi" triggers the display of associated web snippets (Fig. 3) and the web pages.



The accident that experts and utility executives claimed could not happen, did happen at the Fukushima Daiichi Nuclear Power Station in Japan. The accident followed a major earthquake that registered 9.0 on the Richter scale, which in turn triggered a ...

Power plant name (Unit number) Location Reactor type Licensed output (MWe) Start of commercial operation, Fukushima Daiichi(Unit.1) Fukushima prefecture: BWR

Das Kernkraftwerk Fukushima Daiichi, Fukushima Dai-ichi oder Fukushima I ([??????k????ima], jap. ??????????????????????????????, Fukushima Dai-ichi Genshiryoku ...

Workers battling to prevent nuclear meltdown at the stricken Fukushima Daiichi plant were evacuated from the most dangerous areas on Wednesday morning ...

The Fukushima Daiichi Incident ??? Dr. Matthias Braun - 27 March 2011 - p.1. The Fukushima Daiichi Incident 1. 2. 3. 4. 5. Plant Design Accident Progression ...

The accident that experts and utility executives claimed could not happen, did happen at the Fukushima Daiichi Nuclear Power Station in Japan. The accident followed a major earthquake that registered 9.0 on the Richter scale, which in turn triggered a ...

Als Nuklearkatastrophe von Fukushima werden eine Reihe von katastrophalen Unf??llen und schweren St??rf??llen im nordost-japanischen Kernkraftwerk Fukushima Daiichi ...

Fig. 3. The snippets that are associated with the node label "fukushima dai–ichi" of the topic graph from Fig. 2. A single touch on this snippet triggers a call to the iPad web browser in order to display the corresponding web page. In order to go back to the topic graph, the user simply touches the button labeled iGNSSMM on the left upper corner of the iPad screen.

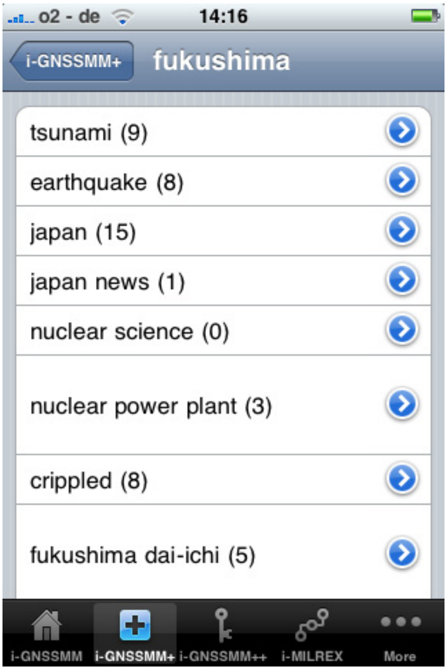
3.2 Text–based User Interface on the iPhone

The next screenshots (Fig. 4 and 5) show the results of the same query displayed on the iPhone.

Fig. 4 shows the alternative representation of the topic graph displayed in Fig. 1. By single touching an item in the list the next page with associated topics to this item is shown. Finally, Fig. 5 presents the snippets after touching the item "fukushima daiichi". Touching one snippet will lead to the corresponding web page.

4 Semantic–Driven Filtering of Extracted Topics

The motivation for using the chunk–pair–distance statistics as described in section 2 is the assumption that the strength of hidden relationships between chunks can be covered by means of their collocation degree and the frequency of their relative positions in sentences extracted from web snippets, and as such, are emphasizing syntactic relationships. In general, chunking crucially depends on the quality of the embedded PoS tagger. However, it is known that PoS tagging performance of even the best taggers decreases substantially when applied on web pages [7]. Web snippets are even harder to process because they are not necessarily contiguous pieces of texts. For example, an



tsunami (9)

earthquake (8)

japan (15)

japan news (1)

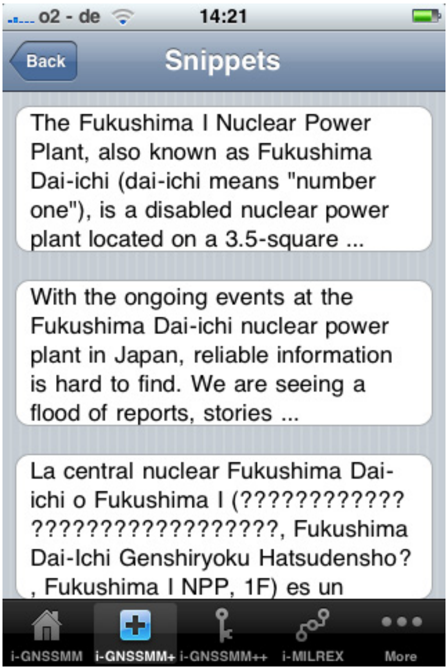
nuclear science (0)

nuclear power plant (3)

crippled (8)

fukushima dai-ichi (5)

Fig. 4. The alternative representation of the topic graph displayed in Fig. 1 on the iPhone



The Fukushima I Nuclear Power Plant, also known as Fukushima Dai-ichi (dai-ichi means "number one"), is a disabled nuclear power plant located on a 3.5-square ...

With the ongoing events at the Fukushima Dai-ichi nuclear power plant in Japan, reliable information is hard to find. We are seeing a flood of reports, stories ...

La central nuclear Fukushima Dai-ichi o Fukushima I (??????????? ?????????????????, Fukushima Dai-Ichi Genshiryoku Hatsudensho? , Fukushima I NPP, 1F) es un

Fig. 5. The snippets after touching the item “fukushima daiichi”

initial manual analysis of a small sample revealed, that the extracted chunks sometimes are either incomplete or simply wrong. Consequently, this also caused the “readability” of the resulting topic graph due to “meaningless” relationships. Note that the decreased quality of PoS tagging is not only caused by the different style of the “snippet language”, but also because PoS taggers are usually trained on linguistically more well–formed sources like newspaper articles (which is also the case for our PoS tagger in use which reports an F–measure of 97.4% on such text style).

Nevertheless, we want to benefit from PoS tagging during chunk recognition in order to be able to identify, on the fly, a shallow phrase structure in web snippets with minimal efforts. In order to tackle this dilemma, investigations into additional semantical–based filtering seems to be a plausible way to go.

About the Performance of Chunking Web Snippets

As an initial phase into this direction we collected three different corpora of web snippets and analysed them according to the amount of well–formed sentences and incomplete sentences contained in the web snippets. Furthermore, we also randomly selected a subset of 100 snippets from each corpus and manually evaluated the quality of the PoS tagging result. The snippet corpora and results of our analysis are as follows (the shortcuts mean: #s = number of snippets retrieved, #sc = well–formed sentences within the set of snippets, #si = incomplete sentences within the snippets, #w = number of words, F(x) = F–measure achieved by the PoS tagger on a subset of 100 snippets with x words).

Fukushima. This corpus represents snippets mainly coming from official online news magazines. The corpus statistics are as follows:

#s	#sc	#si	#w	F(2956)
240	195	182	6770	93.20%

Justin Bieber. This corpus represents snippets coming from celebrity magazines or gossip forums. The corpus statistics are:

#s	#sc	#si	#w	F(3208)
240	250	160	6420	92.08%

New York. This corpus represents snippets coming from different official and private homepages, as well as from news magazines. The corpus statistics are:

#s	#sc	#si	#w	F(3405)
239	318	129	6441	92.39%

This means that 39% of all tagged sentences have been incomplete and that the performance of the Pos tagger decreased by about 5% F–measure (compared to the reported 97.4% on newspaper). Consequently, a number of chunks are incorrectly recognized. For example, it turned out that date expressions are systematically tagged as nouns, so that they will be covered by our noun chunk recognizer although they should not (cf. section 2). Furthermore, the genitive possessive (the "'s" as in "Japan's president") was classified wrongly in a systematic way which also had a negative effect on the performance of the noun chunker. Very often nouns were incorrectly tagged as verbs because of erroneously identified punctuation. Thus, we need a filtering mechanism that is able to identify and remove the wrongly chunked topic–pairs.

Semantic Filtering of Noisy Chunk Pairs

A promising algorithmic solution to this problem is provided by the online clustering system *Carrot2* [14] that computes sensible descriptions of clustered search results (i.e., web documents). The Carrot2 system is based on the Lingo [13] algorithm. Most algorithms for clustering open text follow a kind of "document–comes–first" strategy, where the input documents are clustered first and then, based on these clusters, the descriptive terms or labels of the clusters are determined, cf. [6]. The Lingo algorithm actually reverses this strategy by following a three–step "description–comes–first" strategy (cf. [13] for more details): 1) extraction of frequent terms from the input documents, 2) performing reduction of the (pre–computed) term–document matrix using Singular Value Decomposition (SVD) for the identification of latent structure in the search results, and 3) assignment of relevant documents to the identified labels.

The specific strategy behind the Lingo algorithm matches our needs for finding meaningful semantic filters very well: we basically use step 1) and 2) to compute a set of meaningful labels from the web snippets determined by a standard search engine as described in section 2. According to the underlying latent semantic analysis performed by the Lingo algorithm, we interpret the labels as semantic labels. We then use these labels and match them against the ordered list of chunk–pair–distance elements computed in the topic extraction step described in section 2. This means that all chunk–pair–distance elements that do not have any match with one of the semantic labels are deleted.

The idea is that this filter identifies a semantic relatedness between the labels and the syntactically determined chunks. Since we consider the labels as semantic topics or classes, we assume that the non-filtered pairs correspond to topic–related (via the user query) relevant relationships between semantically related decriptive terms.

Of course, it actually remains to evaluate the quality and usefullness of the extracted topics and topic graph. In the next sections we will discuss two directions: a) a quantitative evaluation against the recognition of different algorithms for identifying named entities and other rigid identifiers, and b) a qualitative evaluation by means of the analysis of user experience.

5 Evaluation of the Extracted Topics

Our topic extraction process is completely unsupervised and web–based, so evaluation against standard gold corpora is not possible, because they simply do not yet exist (or at least, we do not know about them). For that reason we decided to compare the outcome of our topic extraction process with the outcomes of a number of different recognizers for named entities (NEs).

Note that very often the extracted topics correspond to rigid designators or generalized named entities, i.e., instances of proper names (persons, locations, etc.), as well as instances of more fine grained subcategories, such as museum, river, airport, product, event (cf. [11]). So seen, our topic extraction process (abbreviated as *TEP*) can also be considered as a query–driven context–oriented named entity extraction process with the notable restriction that the recognized entities are unclassified. If this perspective makes sense, then it seems plausible to measure the degree of overlap between our topic extraction process and the recognized set of entities of other named entity components to learn about the coverage and quality of *TEP*.

For the evaluation of *TEP* we compared it to the results of four different NE recognizers:

1. SProUT[4]: The *SProUT*–system is a shallow linguistic processor that comes with a rule–based approach for named entity recognition.
2. AlchemyAPI[4]: *AlchemyAPI*–system uses statistical NLP and machine learning algorithms for performing the NE recognition task.
3. Stanford NER[3]: The *Stanford NER*–system uses a character based Maximum Entropy Markov model trained on annotated corpora for extracting NEs.
4. OpenNLP[5]: A collection of natural language processing tools which use the Maxent package to resolve ambiguity, in particular for NE recognition.

We tested all systems with the three snippet corpora described in section 4.

[4] `http://www.AlchemyAPI.com`

[5] `http://incubator.apache.org/opennlp/`

Table 1. Results for query *Justin Bieber*

	I	II	III	IV	V
I	#136	75.64	78.95	78.48	85.37
II	69.01	#143	93.97	86.00	97.17
III	76.71	97.52	#172	92.86	96.09
IV	74.70	89.19	88.52	#196	95.10
V	67.77	79.61	80.66	81.13	#157

Table 2. Results for query *Fukushima*

	I	II	III	IV	V
I	#121	81.03	83.61	81.35	87.5
II	80.26	#129	93.46	87.36	98.48
III	85.00	94.59	#131	91.67	92.22
IV	74.65	89.13	85.26	#178	91.58
V	72.93	80.04	83.19	82.26	#132

Table 3. Results for query *New York*

	I	II	III	IV	V
I	#175	81.39	88.24	85.15	71.05
II	76.60	#169	93.53	86.51	74.36
III	90.00	95.79	#280	92.35	73.28
IV	84.43	92.72	93.17	#230	83.49
V	81.11	83.90	73.77	79.87	#166

Table 4. Summary for NER Evaluation

	I	II	III	IV	V
I	#432	79,25	83.6	81.66	81.31
II	75.29	#441	93.65	86.62	90.00
III	83.90	95.97	#583	92.29	87.19
IV	83.90	95.97	583	#604	87.19
V	73.94	81.18	79.21	81.09	#455

The tables 1, 2, and 3 show the main results for the three different corpora; table 4 shows the results summarised. All numbers denote percentages that show how many relevant[6] NEs of the algorithm in the row could be extracted by the algorithm in the column. For example, in the dataset "Justin Bieber" *TEP* extracted 85.37% of the NEs which have been extracted by *SProUT*. *AlchemyAPI* extracted 75.64% and *StanfordNER* extracted 78.95% of the NEs that have been extracted by *SProUT*. The numbers with preceding "#" show the number of extracted NEs. The following roman numbers are used to denote the different algorithms: I=*SProUT*, II=*AlchemyAPI*, III=*StanfordNER*, IV=*OpenNLP*, and V=*TEP*.

Keeping in mind that our approach always starts with a topic around which all the NEs are grouped, i.e. NE recognition is biased or directed, it is hard to define a gold standard, i.e. manually annotate all NEs which are important in a specific context. In context of the query "Fukushima" most people would agree that word groups describing the nuclear power plant disaster clearly are NEs. Some would also agree that terms like "earthquake" or "tsunami" function as NEs too in this specific context. Given a query like "New York" people probably would not agree that "earthquake" should function as a specific term in this context. Of course there are NEs of generic type like "persons", "locations", or "companies", but it is questionable whether they suffice in the context of our task.

Hence we compared the systems directly with the results they computed. The main interest in our evaluation was whether the extracted NEs by one algorithm can also be extracted by the other algorithms. Furthermore, we set a very simple rating scheme telling us that detected NEs with more occurences are more important than those with lower frequencies.[7]

[6] Relevance here means that a NE must occur more than 4 times in the whole dataset. The value has been experimentally determined.

[7] Except for the *TEP*, where we used the PMI as described above.

The results show that, looking at the numbers and percentages, no system outperforms the others, which on the other hand confirms our approach. Please note that the *TEP* approach works for query-driven context-oriented named entity recognition only. This means that all approaches used in this evaluation clearly have their benefits in other application areas.

Nevertheless by going into details we saw some remarkable differences between the results the systems produced. All systems were able to extract the main general NEs like locations or persons. For terms that are important in the context of actuality and current developments, we saw that the *TEP* approach is able to extract more relevant items. In case of “Fukushima”, the *SProUT* system did not extract terms like “eartquake”, “tsunami” or “nuclear power plant”. Of course this is because the underlying ruleset has not been developed for covering such types of terms. The *AlchemyAPI* and *StanfordNER* systems were able to extract these terms but failed in detecting terms like “accident” or“safety issues”. For “Justin Bieber” relevant items like “movie”, “tourdates” or “girlfriend” could not be detected by all systems except *TEP* . For the snippets associated with the query “New York” all systems identified the most important NEs, and differed for less important NEs only.

Last but not least the runtime, which plays an important role in our system, varied from 0.5 seconds for the *SProUT* system, to 2 seconds for *TEP*, 4 seconds for *StanfordNER* to 15 seconds for *AlchemyAPI*.

6 Evaluation of the Touchable User Interface

For information about the user experience we had 26 testers — 20 for testing the iPad App and 6 for testing the iPhone App: 8 came from our lab and 18 from non–computer science related fields. 15 persons had never used an iPad before, 4 persons have been unfamiliar with smartphones. More than 80 searches have been made with our system and with Google respectively.

After a brief introduction to our system (and the mobile devices), the testers were asked to perform three different searches (using our system on the iPad, iPhone and Google on the iPad/iPhone) by choosing the queries from a set of ten themes. The queries covered definition questions like *EEUU* and *NLF*, questions about persons like *Justin Bieber*, *David Beckham*, *Pete Best*, *Clark Kent*, and *Wendy Carlos* , and general themes like *Brisbane*, *Balancity*, and *Adidas*. The task was not only to get answers on questions like “Who is ...” or “What is ...” but also to acquire knowledge about background facts, news, rumors (gossip) and more interesting facts that come into mind during the search.

Half of the iPad–testers were asked to first use Google and then our system in order to compare the results and the usage on the mobile device. We hoped to get feedback concerning the usability of our approach compared to the well known internet search paradigm. The second half of the iPad–testers used only our system. Here our research focus was to get information on user satisfaction of the search results. The iPhone–testers always used Google and our system mainly because they were fewer people.

Table 5. System on the iPad

#Question	v.good	good	avg.	poor
results first sight	43%	38%	20%	-
query answered	65%	20%	15%	-
interesting facts	62%	24%	10%	4%
suprising facts	66%	15%	13%	6%
overall feeling	54%	28%	14%	4%

Table 6. Google on the iPad

#Question	v.good	good	avg.	poor
results first sight	55%	40%	15%	-
query answered	71%	29%	-	-
interesting facts	33%	33%	33%	-
suprising facts	33%	-	-	66%
overall feeling	33%	50%	17%	4%

Table 7. System on the iPhone

#Question	v.good	good	avg.	poor
results first sight	31%	46%	23%	-
query answered	70%	20%	10%	-
interesting facts	45%	36%	19%	-
suprising facts	56%	22%	11%	11%
overall feeling	25%	67%	8%	-

Table 8. Google on the iPhone

#Question	v.good	good	avg.	poor
results first sight	23%	63%	7%	7%
query answered	70%	20%	10%	-
interesting facts	33%	33%	33%	-
suprising facts	36%	-	27%	37%
overall feeling	25%	33%	33%	9%

After each task, both testers had to rate several statements on a Likert scale and a general questionnaire had to be filled out after completing the entire test. The tables 5, 6, 7, and 8 show the overall result.

The results show that people prefer the result representation and accuracy in the Google style when using the iPad. Especially for the general themes the presentation of web snippets is more convenient and easier to understand. The iPhone–testers could be divided into two groups: in case they were unfamiliar with smartphones the testers preferred our system because it needs much less user interaction and the result are more readable. Testers being familiar with smartphones again prefered the Google style mainly because they are used to it.

However, when it comes to interesting and suprising facts users enjoyed exploring the results using the topic graph (iPad) or the navigation based representation (iPhone/ iPod). The overall feeling was in favor of our system which might also be due to the fact that it is new and somewhat more playful.

The replies to the final questions: *How successful were you from your point of view? What did you like most/least;? What could be improved?* were informative and contained positive feedback. Users felt they had been successful using the system. They liked the paradigm of the explorative search on the iPad and preferred touching the graph instead of reformulating their queries. For the iPhone they prefered the result representation in our system in general and there have been useful comments for improving it. One main issue is the need of a summary or a more knowledge based answer to the search query as Google often does it by offering a direct link to wikipedia as a first search result. This will be part of our future research.

Although all of our test persons make use of standard search engines, most of them can imagine to use our system at least in combination with a search engine on their own mobile devices. The iPhone test group even would use our system as their main search tool (on the smartphone) when the proposed improvements have been implemented.

7 Related Work

Our approach is unique in the sense that it combines interactive topic graph extraction and exploration on different mobile devices with recently developed technology from exploratory search, text mining and information extraction methods. As such, it learns from and shares ideas with other research results. The most relevant ones are briefly discussed below.

Exploratory Search. [10] distinguishes three types of search activities: a) lookup search, b) searching to learn, and c) investigative search, where b) and c) are considered as forms of exploratory search activities. Lookup search corresponds to fact retrieval, where the goal is to find precise results for carefully specified questions with minimal need for examinating and validating the result set. The learn search activity can be found in situations where the found material is used to develop new knowledge and basically involves multiple iterations of search. It is assumed that the returned set of objects maybe instantiated in various media, e.g., graphs, maps or texts. Investigative searching is a next level of search activity that supports investigation into a specific topic of interest. It also involves multiple iterations even for very long periods and the results are usually strictly assessed before they are integrated into knowledge bases. Our proposed approach of exploratory search belongs to the searching to learn activity. In this spirit, our approach is more concerned with recall (maximizing the number of possibly relevant associated topics that are determined) than precision (minimizing the number of possibly irrelevant associated topics that are determined).

Collocation Extraction. We consider the extraction of a topic graph as a specific *empirical collocation extraction task*. However, instead of extracting collocations between words, which is still the dominating approach in collocation extraction research (e.g., [2]), we are extracting collocations between chunks, i.e., word sequences. Furthermore, our measure of association strength takes into account the distance between chunks and combines it with the PMI (pointwise mutual information) approach [15].

[6] also exploit the benefit of Web snippets for improved internet search by grouping the web snippets returned by auxiliary search engines into disjoint labeled clusters. As we do, they also consider methods for automatic labeling. However, their focus is on improving clustering of terms and not on the extraction of empirical collocations between individual terms. Furthermore, they advocate the "document–comes–first" approach of clustering Web snippets which is inappropriate for our methodology, cf. sec. 4.

Unsupervised Information Extraction. Web–based approaches to unsupervised information extraction have been developed by Oren Etzioni and colleagues, cf. [1]; [5]; [16]. They developed a range of systems (e.g., KnowItAll, Textrunner, Resolver) aimed at extracting large collections of facts (e.g., names of scientists or politicians) from the Web in an unsupervised, domain-independent, and scalable manner. They also argue for light–weight NLP technologies and follow a similar approach to chunk extraction as we do (but not a chunk–pair–distance statistics). Although we do not yet explicitly extract relations in the sense of standard relation extraction, our topic graph extraction process together with the clustering mechanism can be extended to also support relation extraction, which will be a focus of our next research.

8 Conclusions and Outlook

We presented an approach of interactive topic graph extraction for exploration of web content. The initial information request is issued online by a user to the system in the form of a query topic description. The topic query is used for constructing an initial topic graph from a set of web snippets returned by a standard search engine. At this point, the topic graph already displays a graph of strongly correlated relevant entities and terms. The user can then request further detailed information through multiple iterations.

A prototype of the system has been realized on the basis of two specialized mobile touchable user interfaces for operation on an iPad and on an iPhone which receive both the same topic graph data structure as input. We believe that our approach of interactive topic graph extraction and exploration, together with its implementation on a mobile device, helps users explore and find new interesting information on topics about which they have only a vague idea or even no idea at all.

Our next future work will consider the integration of open shared knowledge bases into the learn search activity, e.g., Wikipedia or other similar open web knowledge sources and the extraction of relations, and finally to merge information from these different resources. We already have embedded Wikipedia's infoboxes as background knowledge but not yet integrated them into the extracted web topic graphs, cf. [12] for some more details. If so done, we will investigate the role of Wikipedia and the like as a basis for performing disambiguation of the topic graphs. For example, currently, we cannot distinguish the associated topics extracted for a query like "Jim Clark" whether they are about the famous formula one racer or the Netscape founder or even about another person.

In this context, the extraction of semantic relations will be important. Currently, the extracted topic pairs only express certain semantic relatedness, but the nature and meaning of the underlying relationship is unclear. We have begun investigating this problem by extending our approach of chunk–pair–distance extraction to the extraction of triples of chunks with already promising initial results.

Acknowledgements. The presented work was partially supported by grants from the German Federal Ministry of Economics and Technology (BMWi) to the DFKI THESEUS project (FKZ: 01MQ07016).

References

1. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction from the web. In: Proceedings of IJCAI 2007, pp. 2670–2676 (2007)
2. Baroni, M., Evert, S.: Statistical methods for corpus exploitation. In: Lüdeling, A., Kytö, M. (eds.) Corpus Linguistics. An International Handbook. Mouton de Gruyter, Berlin (2008)
3. Dingare, S., Nissim, M., Finkel, J., Grover, C., Manning, C.D.: A system for identifying named entities in biomedical text: How results from two evaluations reflect on both the system and the evaluations. Comparative and Functional Genomics 6, 77–85 (2004)
4. Drozdzynski, W., Krieger, H.-U., Piskorski, J., Schäfer, U., Xu, F.: Shallow processing with unification and typed feature structures — foundations and applications. Künstliche Intelligenz, 17–23 (2004)
5. Etzioni, O.: Machine reading of web text. In: Proceedings of the 4th International Conference on Knowledge Capture, Whistler, BC, Canada, pp. 1–4 (2007)
6. Geraci, F., Pellegrini, M., Maggini, M., Sebastiani, F.: Cluster generation and labeling for web snippets: A fast, accurate hierarchical solution. Journal of Internet Mathematics 4(4), 413–443 (2006)
7. Giesbrecht, E., Evert, S.: Part-of-speech tagging - a solved task? an evaluation of pos taggers for the web as corpus. In: Proceedings of the 5th Web as Corpus Workshop (2009)
8. Gimenez, J., Marquez., L.: Svmtool: A general pos tagger generator based on support vector machines. In: Proceedings of LREC 2004, pp. 43–46 (2004)
9. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to information retrieval. Cambridge University Press (2008)
10. Marchionini, G.: Exploratory search: from finding to understanding. Commun. ACM 49(4), 41–46 (2006)
11. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Journal of Linguisticae Investigationes 30(1), 1–20 (2007)
12. Neumann, G., Schmeier, S.: A mobile touchable application for online topic graph extraction and exploration of web content. In: Proceedings of the ACL-HLT 2011 System Demonstrations (2011)
13. Osinski, S., Stefanowski, J., Weiss, D.: Lingo: Search results clustering algorithm based on singular value decomposition. In: Proceedings of the International IIS: Intelligent Information Processing and Web Mining Conference. Springer (2004)
14. Osinski, S., Weiss, D.: Carrot2: Making sense of the haystack. In: ERCIM News (2008)
15. Turney, P.D.: Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In: Flach, P.A., De Raedt, L. (eds.) ECML 2001. LNCS (LNAI), vol. 2167, pp. 491–502. Springer, Heidelberg (2001)
16. Yates, A.: Information extraction from the web: Techniques and applications. Ph.D. Thesis, University of Washington, Computer Science and Engineering (2007)

A Framework for Interpreting Bridging Anaphora

Parma Nand and Wai Yeap

School of Computing and Mathematical Sciences, Auckland University of Technology,
Auckland, New Zealand
{pnand,yweap}@aut.ac.nz
http://www.aut.ac.nz

Abstract. In this paper we present a novel framework for resolving bridging anaphora. We argue that anaphora, particularly bridging anaphora, is used as a shortcut device similar to the use of compound nouns. Hence, the two natural language usage phenomena would have to be based on the same theoretical framework. We use an existing theory on compound nouns to test its validity for anaphora usages. To do this, we used human annotators to interpret indirect anaphora from naturally occurring discourses. The annotators were asked to classify the relations between anaphor-antecedent pairs into relation types that have been previously used to describe the relations between a modifier and the head noun of a compound noun. We obtained very encouraging results with an average Fleiss's κ value of 0.66 for inter-annotation agreement. The results were evaluated against other similar natural language interpretation annotation experiments and were found to compare well.

In order to determine the prevalence of the proposed set of anaphora relations we did a detailed analysis of a subset 20 newspaper articles. The results obtained from this also indicated that a majority (98%) of the relations could be described by the relations in the framework. The results from this analysis also showed the distribution of the relation types in the genre of news paper article discourses.

Keywords: Anaphora resolution, Noun phrase anaphora, Discourse structure, Noun compounds, Noun phrases.

1 Introduction

The term **anaphora** originated from an ancient Greek word "$\alpha\upsilon\alpha\phi o\rho\alpha$" which means "the act of carrying back upstream". In the context of natural language processing, the term **anaphor** is a reference which points back to a noun that has been mentioned previously in the text being processed. The referred noun is called the **antecedent**. The anaphor can be the same noun as the antecedent, a variation of the noun or a completely different noun. A common form of anaphor is one in which the anaphor is used as a co-reference pointer to the antecedent noun. This is true in the case of pronouns where the pronoun has a one-to-one relation with the antecedent. It is also true in the case of some noun phrases (NPs) where the anaphoric noun directly co-refers to the antecedent (eg. James Smith/Mr Smith). However a noun can also be used as an indirect reference to a previously mentioned noun. As an example consider the following excerpt:

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 131–144, 2013.

1. *John bought a* ***house****. The* ***windows*** *are wooden.*
2. *John was bitten by a* ***snake*** *on the foot. The* ***poison*** *had gone up to the knee by the time ambulance arrived.*

In the example (1) above, the noun **windows** can only be interpreted fully in the context of the noun **house** mentioned in the previous sentence. In this case, the anaphor-noun **windows** is related to the antecedent-noun in an indirect way, different from the one-to-one co-reference type relation. The example in (2) shows another indirect relation between **poison** and **snake**, however it is not the same as the one in (1). NP anaphora resolution studies (e.g. [24,8,27]) treat these indirect relations as a single category and refer to them as *associative* or *bridging* anaphora. In this study we propose a relational framework that distinguishes between the different types of anaphoric relations that can exist between two nouns, one of which represents the anaphor and the second one represents the antecedent. Hence in this framework, the task of anaphora resolution involves identifying the antecedent as well as the type of relation to the antecedent. To distinguish from the previous works, we will use the term *anaphora interpretation*, instead of *anaphora resolution*, where the latter involves only identification of the antecedent.

Since we are also identifying the type of relation, it is possible for an anaphor to have multiple antecedents, related by the same or a different relation. This is a significant departure from the conventional notion of anaphora resolution where an anaphor is resolved to a single, previously mentioned entity. In the case in which the antecedent is also an anaphor, it is assumed to be already resolved, forming a sequential chain. For some NP anaphora this is inadequate. As an illustration, consider the excerpt below:

The robber jumped out of the ***window***$_1$.
The ***house***$_2$ *belonged to Mr Smith.*
The ***window***$_3$ *is thought to have been unlocked.*

If we allow a single resolution relation for an anaphoric NP, then *window*$_3$ would have to be resolved to either *house*$_2$ or *window*$_1$. In either case, a part of the information would not be captured. A common strategy in most studies (eg. [24,8]) is to resolve to the most recent antecedent. In the case of the above excerpt, this would mean that we resolve *window*$_3$ to *house*$_2$ which can be assumed to be already resolved to *window*$_1$. There are two inadequacies in this strategy; firstly the semantic difference between the relation of *window*$_3$ to window$_1$ and *window*$_3$ to *house*$_2$ is approximated by a single co-reference relation, and secondly as a consequence, the direct relation between *window*$_3$ to *window*$_1$ is not captured. In the proposed framework, we will identify both *house*$_2$ and *window*$_1$ as antecedents and interpret each of them with a different relation. It can be argued that this can be overwhelming since we can form a relation even between a pair of very remote entities. However the constraint in our case is that we are only interested in **relations that give rise to anaphoric use of NPs**. The interpretation framework involves **specifying a relation** between an anaphor and the antecedent hence a consequence of this is that an NP can form relations with **more than one antecedent**. This allows us to represent and interpret anaphoric uses of a noun such as *window* to an occurrence of *house* **and** another occurrence of *window*.

Identification of the specific relations in the proposed framework also allows us a richer interpretation of anaphora which are represented by more than one word, that is, compound nouns. In this case, the framework allows us to interpret the modifier-noun with a relation, in addition to the head noun. As a simple example, the compound noun *battle fatigue*, appearing after the clause "*The battle caused fatigue*" has a co-referential relation to the noun *fatigue*, but in addition it also has some semantic relation (identified later as CAUSE) to the noun *battle*.

Hence, there are two novel aspects to this framework for interpreting anaphora. Firstly it identifies a specific relation between the anaphor and its antecedent. Secondly, it also interprets modifiers beyond using them to merely identify the antecedent for the head noun, that is, it interprets them in the same way as the head noun. A consequential effect of this is that an NP can have more then one antecedent. Thus this framework enables us to determine the relational dependence of an anaphoric NP to **all** other NPs in the discourse.

2 Related Works

NP anaphora resolution has received considerably less attention from computational linguists compared to pronominal anaphora even though the proportion of NP anaphora in natural discourses is either comparable to, or more then the proportion of pronominal anaphora. The reason for this seems to stem from the fact that the problem of pronoun resolution is much better defined compared to NP anaphora. This difference in complexity of the problem also explains why whatever published work is available on NP anaphora resolution, is predominantly focussed on NPs that are definite descriptions (eg. [24,8,2,3]) with the accompanying task of identifying whether a definite NP is anaphoric or not. NP resolution in these studies involves identifying a single previously mentioned noun that the anaphoric NP refers to. Anaphora in these studies have been studied as two categories; *direct* and *associative*. The direct category includes cases in which an NP directly co-refers to another entity such as the case of *he/John*. The associative category includes cases such as *window/house*. Some of the studies such as [24] have gone a step further to specify the actual associative relation in terms of synonymy[1], hyponymy[2] and meronymy[3]. The motivation for these relations seems to have risen from organization of the lexicon, WordNet [7] which is used to bridge the meanings between the anaphor and the antecedent.

In this paper we propose a framework that presents an enhanced interpretation of the generic bridging relations. The framework is based on recognizing that anaphora is used in a way similar to another natural language phenomenon, namely compound noun generation. A compound NP of the form *noun* + *noun* (N + N) consists of two nouns which have some underlying semantic relations ([17,6,19]). According to these studies, use of compound NPs is highly *productive* rather then *lexical*. In this productive process, compound NPs are formed on the fly as a discourse is being produced, rather then recalled and used from a lexicon. In this productive process, the semantic relation

[1] Same meaning relation.

[2] Same subset/superset relation.

[3] Part/whole relation.

between the nouns is deleted and a shortcut is formed by juxtaposing the two nouns to form a compound noun. However, for interpretation of the compound noun the semantic relation is expected to be reconstructed by the consumer ([17]). This process of compound noun generation has been described as *predicate deletion* in literature. The framework proposed in this paper is based on the premise that associative anaphora usage is a similar natural language phenomenon to compound noun generation. They both involve two nouns connected by a relation, but the relation is not explicitly expressed by the producer, rather, it is expected to be deduced by the consumer. The difference is that, in the case of anaphora, the two nouns are used separately as anaphor and antecedent, while in the case of generation of compound nouns, the two nouns are juxtaposed together as a compound noun. Research on the generation of compound nouns is at an advanced stage with various theories existing on how compound nouns are formed. According to these theories, formation of NPs is not totally unconstrained, in other words, a compound noun cannot be formed with any two random nouns. For example, *war man* can not be formed on the basis of the relation "man who hates war" or similarly *house tree* can not be formed from "tree between two houses" [29]. In both the examples there does exist a relation between the nouns, however it is of the type that can be used to form a compound NP. Linguistic studies on compound nouns (eg. [6,29,17,28] have assumed that the set of generic relations are finite and characterizable, although the set is not necessarily common among all the studies. Studies such as [17] and [6] have attempted to identify these relations, and even though the exact set of relations proposed by the different studies are slightly different, a core set is very similar. An additional aspect highlighted in [6] is that compound nouns can also be formed from "temporary or fortuitous" relations, hence it presents a case for existence of unbounded number of relations although the vast majority of compound nouns fit into a relatively small set of categories [26].

The relational frameworks used in computational linguistics vary along similar lines as those proposed by linguists. Some works in the computational linguistics (eg. [4,20]) assume the existence of an unbounded number of relations while others (eg. [16,13]) use categories similar to Levi's finite set. Yet others (eg. [22,14]) are somewhat similar to [28]. Most of the research to date has been domain independent, done on generic corpus such as Penn Tree Bank, British National Corpus or the web.

The later works on noun compounds have followed on from either [18] or [28] with some of them coming up with a slightly different variation while others have defined a finer grained set of relations dictated by the data sets used for the study. For example, [26] reports a set of 43 relations grouped into 10 upper level categories. Most of the relations from different studies can be mapped to an equivalent relation in other studies.

For this study we chose the set of relations proposed in [18] for two reasons. Firstly, our analysis of corpus for anaphor-antecedent relations seemed to map better to Levi's set of nine relations for compound nouns and secondly more of these relations can be computationally determined from existing lexicons such as WordNet and the Web. There are already several works that extract Levi's set of relations from WordNet and the Web with various levels of success. In terms of natural language processing, a linguistic theory is only useful if it can be reasonably implemented in a computational system. The theory on anaphora proposed in this paper can be easily implemented by adopting the relation extraction techniques from compound noun generation works. We are currently in the process of developing an anaphora resolution system by integrating the various relation extraction strategies described in computational works on compound nouns.

3 Anaphora Resolution Framework

In the Introduction we stated that anaphora interpretation and noun compound generation are two indicants of the same underlying relational framework between entities. Hence, a framework describing compound noun generation has to apply to anaphora usage as well. In the proposed framework we extend the relations proposed for compound noun generation from [18] for interpretation of noun phrase as well as pronominal anaphora.

An indirect reference such as *window* referring to *house* and *diesel* referring to *truck* is based on the predicates “house **has** windows” and “a truck **uses** diesel”. In the case of compound noun generation, the predicate is deleted and the two entities are juxtaposed to form the noun compounds *house window* and *diesel truck*. For interpretation of the compound noun the consumer is expected to reconstruct the relation between the modifier and the head noun ([6,18]). We propose that the compound noun generation process is very similar to associative anaphora, except in the latter case the modifier is not necessarily bound to the head noun as part of a noun compound. That is, it may exist in another clause, however the same relation is still expected to be reconstructed for a full interpretation of the anaphor. Hence, for the example for the predicate “house has window”, we could have the full NP, *house window* produced by predicate deletion. However in addition, the same predicate could also be expressed anaphorically as in the following example:

> *John bought a* ***house*** *in Glen Eden.*
> *The* ***windows*** *are wooden.*

In the example above the related entities from the predicate “house has windows” are separated into two different sentences, each expressing information about “house” and “windows” respectively. In order to relate the two sentences we need to bridge the “semantic gap” between “windows” and “house”. This is referred to as *text cohesion* and/or *coherence* ([12,25]). Hence identifying the specific relation between an anaphor and its antecedent is necessary for establishing coherence which is fundamental for a full interpretation of any text.

Semantic relations between certain entities exist by default and can be assumed as part of the lexical knowledge of the consumer. For example, the HAVE relation between *car* and *tyre* is part of lexicon so the noun compound *car tyre* and the noun *tyre* used anaphorically to refer to *car* is readily understood. In addition, the HAVE relation can also be established temporarily in a discourse followed by its use for anaphora and/or compound nouns. For example, after specifying the relation “the box has tyres”, the noun *tyres* can be used to indirectly refer to *box* in the same way as the reference of *tyres* to *car*. However, the former can only be used in the context of the discourse in which the relation was expressed. This corresponds to Downing’s [6] fortuitous relations.

We distinguish between these two type of relations as **persistent** or **contextual**. Persistent relations are those that form part of the lexical knowledge which are valid within the context of a particular discourse as well as all other discourses. On the other hand contextual relations are transient, and may be valid only for the duration of a single discourse, for example, "a cup on a table" or "John has a knife". The contextual relations are expressed as either a verb or a preposition, relating two entities in the discourse being processed. In order to resolve all bridging anaphora in a discourse, we need to identify both persistent as well as the contextual relations. The persistent relations can be expressed either explicitly or assumed as part of the lexicon. On the other hand, the contextual relations have to be expressed explicitly via verbs and prepositions. The question now is which verbs and prepositions represent the anaphoric relations.

As argued earlier, the semantic relations used by bridging anaphora are the same as those used for compound noun generation, hence for this study we adopted the set proposed in [18]. The set of relations consist of CAUSE, HAVE, MAKE, USE, BE, IN, FOR, FROM and ABOUT. In order to define a complete framework for anaphora interpretation, we needed to do two modifications to the nine relations from [18]. Both of these modifications were done in order to be able to better interpret and represent plural anaphoric nouns. This was done by introducing a new relation named ACTION, and by splitting the existing BE relation into BE-INST and BE-OCCR. These are explained next.

When two or more entities in a discourse are participating in the same or similar event, they can be referred to as a unit by a collective NP in the context of the discourse. The entities in the same or similar action can be expressed by the conjunction *and* or described by two different clauses. For instance in the sentence "The coastguard and Lion Foundation Rescue helicopter were called out.", the entities *coastguard* and *Lion Foundation Rescue helicopter* are related to each other by the virtue of participating in the same action. Similarly, the clauses "the truck rolled down the hill" and "the ball rolled down the hill" would enforce the same relation between *truck* and *ball* since they are both engaged in the same action (roll). This relation between truck and ball is only valid for the context of the discourse, hence this relation is contextual. We describe this contextual relation as the ACTION relation which relates entities participating in events which are identified to be same or similar. The ACTION is used to describe an NP such as *runners* used to refer to *fox* and *Peter* from the context clause "The fox and Peter were running".

The second modification involved defining a finer grained BE relation in order to interpret existence of plurals in a different form. We split Levi's BE relation into BE-OCCR and BE-INST to distinguish between direct co-reference or identity relation and an instance relation. In a BE-OCCR relation an NP directly forms a one-to-one co-reference to another NP, eg. *John/he* and *John/the driver*. The BE-INST relation represents cases where an anaphor refers to a plural antecedent, in a partial capacity, for example, *both trucks/northbound truck*. In this case the NP *northbound truck* is an instance of *both trucks* which is distinct from a co-reference relation. It can be argued that all subset/ superset relations such as *John/driver*(John is an instance of driver) and *car/vehicle* (car is and instance of vehicle) is an instance relation. However we consider these as BE-OCCR relation since they **function** to identify the entity. Hence in the framework, the BE-INST relation only relates a plural NP and an NP representing a subset of the plural NP.

With this discussion we can now define and exemplify the eleven relation types used in the anaphora interpretation framework. They are:

CAUSE - Includes all causal relations. For example, *battle/fatigue*, *earthquake/tsunami*

HAVE - Includes notions of possession. This includes diverse examples such as *snake/poison*, *house/window* and *cake/apple*.

MAKE - Includes examples such as *concrete house*, *tar/road* and *lead/pencil.*

USE - Some examples are *drill/electricity* and *steam/ship*.

BE-INST - Includes plural cases such as *both trucks/southbound truck*, *John/teachers*.

BE-OCCR - Describes the same instance participating in multiple events. For example *John Smith/Mr Smith/he* and *John Smith/the driver*.

IN - This relation captures grouping of things that share physical or temporal properties. For example *lamp/table* and *Auckland/New Zealand*.

FOR - This includes purpose of one entity for another. For example *pen/writing* and *soccer ball/play*.

FROM - This includes cases where one entity is derived from another. For example *olive/oil* and *wheat/flour*.

ABOUT - Describes cases where one entity is a topic of the other. For example *travel/story* and *loan/terms*.

ACTION - This is only a contextual relation meant to capture entities engaged in same or similar action either with the same object/s or a null object.

The next section describes the annotation experiment done in order to validate that anaphora usage is based on the above relation types.

4 Annotation Experiment

4.1 Annotators

For the purpose of human validation of all relations in the framework we used second and third year students enrolled in computer related degrees. The annotation experiments were done over a period of 4 weeks at the beginning of their usual classes. Four different streams were used each consisting of approximately 30 students. The students in each stream were given a basic training on the requirements of the annotation and they were given a single annotation task at the beginning of their class over a 4 week period. The whole annotation experiment was broken down into session based tasks involving 25 anaphoric NPs per task. This was done to ensure that each task was completed in about 10 minutes with minimal impact on the students class time. In addition, the annotators were not identified in any of the tasks. We only ensured that an annotator did not annotate the same task twice.

4.2 Annotation Data

Our base input data used for content analysis for all aspects of NP usage consisted of 120 articles (of mixed genre) from *The New Zealand Herald*, *The Dominion Post* and *The Press* which are three major online newspapers from three different cities in New Zealand. The choice of the articles were not completely random. This corpora was developed to serve as the input data for the anaphora resolution system which is the parent project of this study. Hence, the corpora was developed from the articles which were not too short (had more then 20 sentences), exhibited use of a variety of anaphoric uses (including pronominal anaphora) and had been written by different writers.

An inherent challenge in most NLP tasks is what is referred to as *data sparseness*. The term is used to describe a characteristic when a single chosen corpus cannot be used for consistent empirical validation of **all** aspects of a theory. This is because the prevalence of the different characteristics of an NLP theory can be unevenly distributed in a fixed corpus. Hence, we searched an extended corpus in order to make a lower threshold of 15 relations from each category. For this we used The Corpus of Contemporary American English [5]. This freely available corpora consisting of some 410 million words from a variety of genre and has an online web interface which can be used to do fairly complex searches for words and phrases hence forms an excellent resource for manual content analysis for NLP tasks.

We excluded validating the BE-OCCR relation since this is a non-ambiguous co-reference relation.

For the annotation experiment we used 3 streams of approximately 30 students giving us a total of 90 different annotators. Each annotator took 4 different tasks, one per week for a period of 4 weeks. Each task consisted of 25 antecedent-anaphor pairs and was annotated by 2 streams, ie. approx. 60 annotators. We randomly discarded some annotation task sheets in order to have a consistent number of annotations for each pair resulting in 25 annotators for each task. Each relation type from (CAUSE, HAVE, MAKE, USE, BE-INST, IN, FOR, FROM, ABOUT, ACTION) as classified by the author was represented by 15 anaphor-antecedent pairs. The pairs from each of the 10 relation types were randomly selected to make up 6 task sheets, each consisting of 25 pairs. The total number of classifications for all relations amounted to 3750 with 375 classifications for each relation type consisting of 15 different anaphor-antecedent pairs.

Each of the streams were given a basic training on semantic interpretation of the relation types using the examples in section 3. These examples were also given as a separate sheet with each annotation task. Each task sheet consisted of anaphor-antecedent pairs and a tick box for each of the relations. The annotators were asked to choose the relation which best describes the anaphor-antecedent pair. Two additional options, OTHER and NONE were also given. The OTHER was to be used if the annotator thought that a relation does exist but is not present in the given list and option NONE to be used if the annotator thought that the pair were not related at all.

Table 1 shows the confusion matrix of the relation types as identified by the annotators against the author's classification. Table 2 shows the corresponding confusion indices between the relation types. The confusion indices indicate the likelihood of a relation type to be interpreted as another type.

Table 1. Confusion Matrix for the non-normalized NPs. The columns give the annotations by annotators against the author's annotations on the rows. Each relation category had a total of 375 annotations done by 50 different annotators. The bolded entries indicate number of annotations agreeing with author's annotations.

	CAUSE	HAVE	MAKE	USE	BE–INST	IN	FOR	FROM	ABOUT	ACTION	OTHER	NONE
CAUSE	**208**		45	87				4	14		15	2
HAVE		**196**	113	7		13		46				
MAKE		45	**206**	120				4				
USE		45	26	**242**			59	3				
BE-INST					**347**			19			9	
IN		64	37			**241**		33				
FOR		18	5	132			**216**	4				
FROM		9	17			35	87	**227**				
ABOUT	48	11		56				7	**253**			
ACTION									5	**351**	19	

Table 2. Confusion Index Matrix between the relation types corresponding to table 1. The bolded entries indicate the index of annotations which were the same as that of the author's.

	CAUSE	HAVE	MAKE	USE	BE–INST	IN	FOR	FROM	ABOUT	ACTION	OTHER	NONE
CAUSE	**0.55**		0.12	0.23				0.01	0.04		0.04	0.01
HAVE		**0.52**	0.30	0.02		0.03		0.12				
MAKE		0.12	**0.55**	0.32				0.01				
USE		0.12	0.07	**0.65**			0.16	0.01				
BE-INST					**0.93**			0.05			0.02	
IN		0.17	0.10			**0.64**		0.09				
FOR		0.05	0.01	0.35			**0.58**	0.01				
FROM		0.02	0.05			0.09	0.23	**0.61**				
ABOUT	0.13	0.03		0.15				0.02	**0.67**			
ACTION								0.00	0.01	**0.94**	0.05	

4.3 Annotation Results

The first observation of the annotation results from table 1 is that only 2 annotations out of a total of 3750 were classified as NONE indicating that the annotators by and large thought that the pairs given had **some** relation. In addition a total of 43 (approx. 1.1%) annotations were classified as OTHER, which represents the relations which were described by a relation not in the list of 10 that were given. The main categories that were interpreted as having some other relation were CAUSE and ACTION, however these were still a very small percentage with indices of 0.04 and 0.05 respectively. The bolded entries in table 2 give the percentage agreement of the relation types agreeing with that of the author. The relation types BE-INST and ACTION have the highest conformance indicating they are the least ambiguous. The other types vary from a low figure of 0.52 for HAVE to 0.67 for ABOUT, with an overall agreement value of 0.66. The relation types that were easily confused and hence can be interpreted as semantically close, were HAVE, MAKE and USE. Conflating these 3 categories gives us an agreement index of 0.89. Another crucial observation is for the FROM relation. Although not by large amounts, this relation type seems to be confused with all other categories. This prompted us to closely examine the task sheets to see if there were

Table 3. Inter-annotator agreement comparison between studies dealing with relations between composite nouns of noun compounds

Study	Agreement Index	No. of Relations
[26]	0.57 - 0.67 κ	43
[9]	0.61 κ	22
[23]	0.68 κ	6
[14]	52.31 %	20
[10]	0.58 κ	21

consistent misclassifications by the author, however no such patterns were found. Some of the classifications seemed to use the FROM type as a "fall back" category.

In order to compare the inter-annotator agreement with other similar studies we also computed the Fleiss' κ measure. The κ index for the overall annotation tasks was computed to be 0.64 and the value with HAVE, MAKE and USE conflated was 0.86. The overall κ value 0.64 compares well with the inter-annotator figures from other annotation experiments dealing with identification of relations. For comparison some of the results are summarized in table 3.

5 Prevalence of Anaphoric Relations in News Articles

In order to gauge the prevalence of the anaphoric relations in naturally occurring discourses, we used 30 of the 120 newspaper articles used for the annotation experiment and analyzed them in detail to determine the existence and the distribution of the proposed relations. The set of 30 articles consisted of 352 sentences and 2323 nouns. The 30 randomly chosen articles were analyzed by the author for the existence of the 10 relations from table 1. In addition there were two additional relation types. The first one was the OTHER relation for relations that could not be categorized into any of the 10 types from table 1. The second one was the BE-OCCR relation which was considered trivial, hence was not tested in the annotation experiment. This represents the identity or the co-reference relation.

Out of a total of 2323 nouns, 1324, or 57% were found to be used anaphorically. This shows that more than half of the nouns used were anaphoric, hence highlights the importance of being able to resolve them for discourse interpretation. Note that in our framework, an anaphor can have more than one antecedent where the antecedents are related by different relations. The 1324 nouns used anaphorically had a total of 1588 relations between them. This gives us an average of 1.2 relations per anaphor. The detailed distribution of the relation types are shown in figure 1. The figure firstly shows that the majority (524) of the relations are of type BE-OCCR which are identity relations represented by both pronouns as well as noun phrases. The reset of the relations were fairly evenly distributed ranging from 64 to 175. Only a small number, 32 or 2% of the relations were found to be outside the range of the relations in the framework. Aside from the BE-OCCR and OTHER relation types, there were 1032 bridging relation from the list BE-INST, CAUSE, HAVE, MAKE, USE, IN, FOR, FROM, ABOUT and ACTION. This means a substantial proportion (65%) of relations were bridging, highlighting their prevalence in news paper articles. We are in the process of implementing the resolution of these types of bridging as well as the traditional co-reference anaphora at a discourse level. The resulting network of relations between nouns in the discourse will provide us with an infrastructure which can be utilized in a computational system for a richer interpretation of discourses.

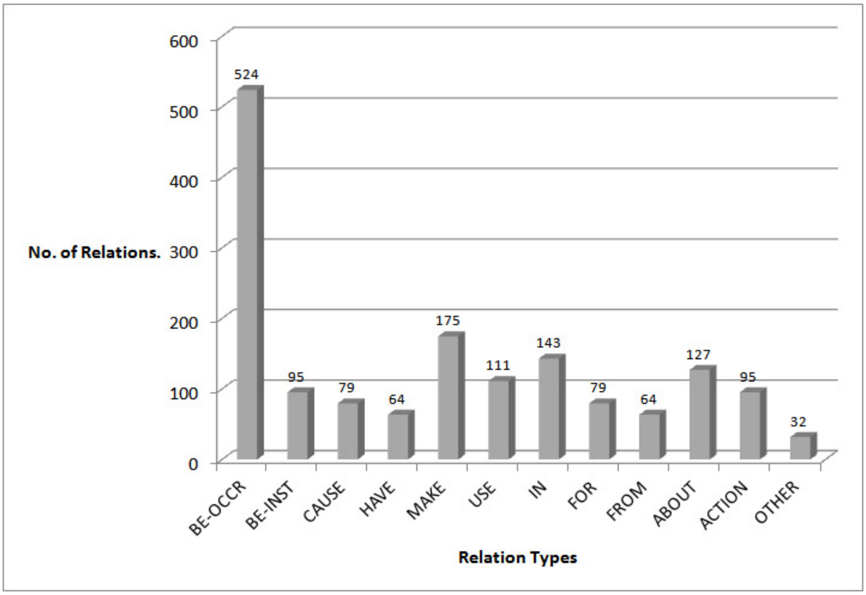


Fig. 1. Figure showing the relative distribution of relation types in a corpus of 20 news paper articles

6 Discussion

The annotation experiment results strongly indicate that the two natural language usage phenomena of compound noun generation and anaphoric use of nouns are based on the same underlying semantic structure. At a theoretical level, this has a significant impact on our understanding of how humans use natural language. In particular, it will help us better understand the use of compound nouns which are also anaphoric by using the same theory to interpret them. At a computational level, the proposed framework for anaphora resolution allows us to marry the two nlp areas so that we can better share computational advances in the two research areas. Recently there has been an increased momentum [13,26,20,4,15,21,1,11] towards automatic derivation of relations between composite nouns in noun compounds, most of them based on relations from [18]. This will result in an increasing amount of ontology representing semantic relations used for generating compound nouns. Any such ontology will be directly useful for anaphora resolution in the framework proposed in this paper.

Another significant advantage of a common framework is that it will be easier to integrate the full meaning of a compound noun and the meaning associated with it being used anaphorically. Currently, anaphora is described using a different set of relations (eg. synonymy, hypernomy, meronomy etc.) and compound nouns with a different set. Hence, when interpreting a compound noun which is also anaphoric, it becomes difficult to merge the two meanings. Combining the processes within the same framework gives us a much stronger interpretative power enabling us to interpret a modifier as well as the head noun. As an illustration consider the excerpt below:

> *John's **car** had an accident yesterday. Its thought **faulty car tyres** played a major role in the accident.*

The compound noun *faulty car tyres* expresses the relation HAVE between the modifier *car* and the head noun *tyres*, defined by the compound noun generation framework. In terms of straight forward anaphora resolution, the compound noun *faulty car tyres* is not anaphoric since the head noun *tyres* does not co-refer to anything in the previous sentence. However to be able to fully interpret the meaning of the second sentence, it is crucial that we know that the noun *car* in the first sentence also has a HAVE relation to *tyres* in the second sentence. This relation forms the basis of the coherence between "car and accident" in the first sentence and "tyres and accident" in the second sentence. The proposed framework enables us to use relations from the same set to describe the relations between *car* and *tyres* in the compound noun *faulty car tyres* and the anaphoric relations between *faulty car tyres* and *car*. The resultant output from processing a whole discourse using the proposed framework would be a network of entity-to-entity relations consisting of all freely existing nouns as well as nouns participating as modifiers. This network can either be used on its own or used as a building block towards higher level discourse structures such as a coherence structure.

7 Concluding Remarks

In this paper we presented a relational framework for interpreting anaphoric NPs which goes beyond the conventional co-reference relations. We argued that anaphora usage and compound noun generation are based on a common relational framework. To support this we used an existing NP production framework and validated it for anaphora usage using real world discourses. We also argued that by using this framework, a more accurate level of discourse interpretation can be achieved, both directly, as well as using it as a building block for a higher level discourse structure such as the coherence structure. We are in the process of implementing the framework and will be reporting the results in near future. It is anticipated that successful computation of this framework will help in numerous NLP tasks such as document visualization, summarization, archieving/retrieval and search engine applications.

References

1. Barker, K., Szpakowicz, S.: Semi-automatic recognition of noun modifier relationships (1998)
2. Bean, D., Riloff, E.: Corpus-based identification of non-anaphoric noun phrases. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, pp. 373–380. Association for Computational Linguistics, Morristown (1999)

3. Bunescu, R.: Associative anaphora resolution: A web-based approach. In: Proceedings of the EACL 2003 Workshop on the Computational Treatment of Anaphora, pp. 47–52 (2003)
4. Butnariu, C., Veale, T.: A concept-centered approach to noun-compound interpretation. In: Proceedings of the 22nd International Conference on Computational Linguistics, COLING 2008, vol. 1, pp. 81–88. Association for Computational Linguistics, Stroudsburg (2008), http://portal.acm.org/citation.cfm?id=1599081.1599092
5. Davies, M.: Corpus of contemporary american english (2010), http://www.americancorpus.org
6. Downing, P.: On the creation and use of english compound nouns. Language 53(4), 810–842 (1977)
7. Fellbaum, C.: WordNet: An Electronic Lexical Database. Bradford Books (1998)
8. Fraurud, K.: Definiteness and the processing of noun phrases in natural discourse. Journal of Semantics 7(4), 395–433 (1990)
9. Girju, R.: Improving the interpretation of noun phrases with crosslinguistic information. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp. 568–575 (2007)
10. Girju, R., Moldovan, D., Tatu, M., Antohe, D.: On the semantics of noun compounds. Computer Speech and Language 19(4), 479–496 (2005)
11. Hendrickx, I., Kim, S.N., Kozareva, Z., Nakov, P., Séaghdha, D.O., Padó, S., Pennacchiotti, M., Romano, L., Szpakowicz, S.: Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In: Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval 2010, pp. 33–38. Association for Computational Linguistics, Stroudsburg (2010), http://portal.acm.org/citation.cfm?id=1859664.1859670
12. Hobbs, J.R.: Coherence and coreference. Cognitive Science 67, 67–90 (1979)
13. Kim, S.N., Nakov, P.: Large-scale noun compound interpretation using bootstrapping and the web as a corpus. In: The Proceeding of Conference on Empirical Methods in Natural Language Processing, EMNLP, Edinburgh, UK (2011)
14. Kim, S.-N., Baldwin, T.: Automatic Interpretation of Noun Compounds Using WordNet Similarity. In: Dale, R., Wong, K.-F., Su, J., Kwong, O.Y. (eds.) IJCNLP 2005. LNCS (LNAI), vol. 3651, pp. 945–956. Springer, Heidelberg (2005)
15. Kim, S.N., Baldwin, T.: Interpreting semantic relations in noun compounds via verb semantics. In: Proceedings of the COLING/ACL on Main Conference Poster Sessions, COLING-ACL 2006, pp. 491–498. Association for Computational Linguistics, Stroudsburg (2006), http://portal.acm.org/citation.cfm?id=1273073.1273137
16. Lauer, M.: Corpus statistics meet the noun compound: some empirical results. In: Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics, ACL 1995, pp. 47–54. Association for Computational Linguistics, Stroudsburg (1995), http://dx.doi.org/10.3115/981658.981665
17. Levi, J.: Where do all those other adjectives come from. Chicago Linguistic Society 9, 332–354 (1973)
18. Levi, J.N.: The syntax and semantics of complex nominals. Academic Press, New York (1978)
19. Li, C.N.: Semantics and the Structure of Compounds in Chinese. Ph.D. thesis, University of Carlifornia dissertation (1971)
20. Nakov, P.: Noun compound interpretation using paraphrasing verbs: Feasibility study (2008), http://www.cs.berkeley.edu/~nakov/selected_papers_list/aimsa2008.pdf
21. Nastase, V., Sayyad-Shiarabad, J., Sokolova, M., Szpakowicz, S.: Learning noun-modifier semantic relations with corpus-based and wordnet-based features. In: Proceedings of the National Conference on Aritficial Intelligence, vol. 21(Part 1), pp. 781–787 (2006)

22. Nastase, V., Szpakowicz, S.: Exploring noun-modifier semantic relations. In: Proceedings of the 5th International Workshop on Computational Semantics (2003)
23. Ó Séaghdha, D.: Annotating and learning compound noun semantics. In: Proceedings of the 45th Annual Meeting of the ACL: Student Research Workshop, ACL 2007, pp. 73–78. Association for Computational Linguistics, Stroudsburg (2007), http://portal.acm.org/citation.cfm?id=1557835.1557852
24. Poesio, M., Vieira, R., Teufel, S.: Resolving bridging references in unrestricted text. In: Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts, ANARESOLUTION 1997, pp. 1–6. Association for Computational Linguistics, Morristown (1997), http://portal.acm.org/linebreakcitation.cfm?id=1598819.1598820
25. Sanders, T.: Toward a taxonomy of coherence relations. Discourse Processes 15(1), 1–35 (1992)
26. Tratz, S., Hovy, E.: A taxonomy, dataset, and classifier for automatic noun compound interpretation. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL 2010, pp. 678–687. Association for Computational Linguistics, Stroudsburg (2010), http://portal.acm.org/citation.cfm?id=1858681.1858751
27. Vieira, R., Poesio, M.: An empirically based system for processing definite descriptions. Computational Linguistics 26(4), 539–593 (2000), http://www.mitpressjournals.org/doi/abs/10.1162/089120100750105948
28. Warren, B.: Semantic patterns of noun-noun compounds. Gothenburg studies in English. Acta Universitatis thoburgensis (1978)
29. Zimmer, K.E.: Some general observations about nominal compounds. Working Papers on Language Universals, pp. C1–C21 (1971)

Toward Information Sharing of Natural Disaster: Proposal of Timeline Action Network

The-Minh Nguyen, Takahiro Kawamura, Yasuyuki Tahara, and Akihiko Ohsuga

Graduate School of Information Systems, The University of Electro-Communications,
1-5-1, Chofugaoka, Chofu-shi, Tokyo, 182-8585, Japan
{minh,kawamura,tahara,akihiko}@ohsuga.is.uec.ac.jp

Abstract. In emergency situations such as earthquake, typhoon, it is important to share people's actions in real-time. Therefore, in this paper, we first design a timeline action network based on Web Ontology Language (OWL) in order to represent these actions in real-time. We then use our previous work to automatically collect data for the action network from Twitter. Finally, we propose a novel action-based collaborative filtering, which predicts missing activity data, in order to complement this timeline action network. Moreover, with a combination of collaborative filtering and natural language processing (NLP), our method can deal with minority actions such as successful actions. Based on evaluation of tweets which related to the massive Tohoku earthquake, we indicated that our timeline action network can share useful action patterns in real-time. Not only earthquake disaster, our research can also be applied to other disasters and business models, such as typhoon, travel, marketing, etc.

Keywords: Evacuation-rescue, Twitter, Action network, Action-based Collaborative Filtering.

1 Introduction

The ability of computers to recommend useful action patterns based on users' behaviors is now an important issue in context-aware computing [1], ubiquitous computing [2], and can be applied to assist people in disaster areas. When the massive Tohoku earthquake and Fukushima nuclear disaster occurred in March 2011, many people felt panic, and wanted to know "what did other people do to go to a safe place", "how to come back home", etc. Since the train system in Tokyo was stopped after the earthquake, there are about 3 million people in Tokyo who had difficulties returning home [3]. The Japanese government said that there is 87% of chance of an approximately 8.0-magnitude earthquake occurring in the Tokai region within the next 30 years [4]. In this case, temporary homeless people who are unable to return home, are expected to reach an amount of 6.5 million [5].

To recommend useful action patterns, it is necessary to have a *collective intelligence of these action patterns*. During the massive Tohoku earthquake, while landlines and mobile phones got stuck, Twitter was used to exchange information related to evacuation, traffic, etc. On 11 March, 2011 the number of tweets from Japan dramatically increased to about 33 million [6], 1.8 times higher than the average figure. Therefore,

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 145–157, 2013.

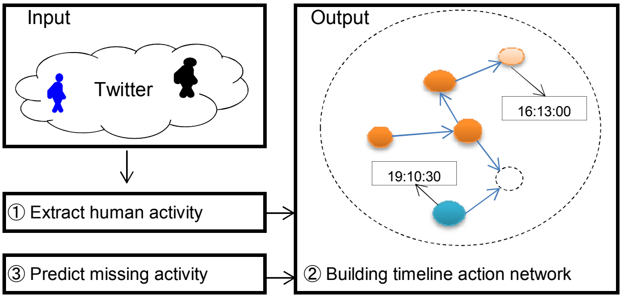

Input
Twitter
Output
16:13:00
19:10:30
① Extract human activity
③ Predict missing activity
② Building timeline action network

Fig. 1. Approach of building timeline action network

we can say that Twitter is becoming the *sensor* of the real world. In other words, we can collect activities of people in earthquake disaster from Twitter. We can consider these activities as the collective intelligence. However, sentences retrieved from Twitter which are more complex than other media texts, are often structurally various, syntactically incorrect, and have many user-defined new words [7].

In emergency situations, it is important to let the computers make a useful recommendation *in time*. This means that the activities should be collected, and represented in real-time. However, since tweets depend on users' autonomy, it is highly probable that the users do not post their activities in real-time. Thus, we need to solve the problem of missing activity data in order to have activity data in real-time. Additionally, to help the computer understand the meaning of the data, we should build the collective intelligence based on OWL.

In this paper, as shown in Figure 1, we first use our previous work [7] to automatically extracts human activities from Twitter. And then, we design a timeline action network (TiAN) to represent these activities in real-time. Finally, we propose a novel action-based collaborative filtering, which predicts missing activity data, to complement the action network. Moreover, with a *combination of collaborative filtering and NLP*, our method can handle *minority actions* such as successful actions.

The main contributions of our work are summarized as follows:

- We has successfully designed the TiAN based on OWL in order to represent the activities in real-time.
- We also proposed a method that can predict missing activity data to complement the action network. Moreover, this method can handle *minority actions*.

The remainder of this paper is organized as follows. We first explain how to extract human activity from Twitter. Secondly, we design TiAN to represent the extracted activities. Thirdly, we explain how to predict missing activity data. We then report our experimental results, and explain how to apply the action network. After considering related works, this paper concludes with some discussions of the future work.

2 Extraction of Human Activity

In this paper, we define an activity by five attributes namely *actor*, *act*, *object*, *time* and *location*. And an *action* consists of a combination of *act* with *object*. For example, in

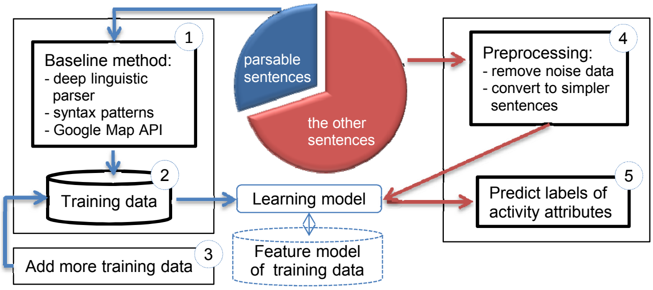


Fig. 2. Approach of extracting activities from Twitter

the sentence “Tanaka is now taking refuge at Akihabara”, *actor*, *act*, *time* and *location* are “Tanaka”, “take refuge”, “now”, “Akihabara” respectively. Figure 2 explains the key ideas of our previous work [7] for extracting activity attributes in each sentence retrieved from Twitter, are summarized as follows:

- We deploy self-supervised learning [8], and use the linear-chain conditional random field (CRF) [9] as a learning model.
- Using deep linguistic parser [10], 8 syntax patterns, and Google Map API to make initial training data from parsable sentences.
- Based on the initial training, we collect more trustworthy training from weblogs.
- Since sentences retrieved from Twitter often contain noise data, we remove these noise data before testing. Additionally, to avoid error when testing, we convert complex sentences to simpler sentences by simplifying noun phrases and verb phrases.
- We consider not only time stamp of tweets, but also time expression (now, this evening, etc) to decide time of activities in these tweets.

3 Building Timeline Action Network

In this section, we first design TiAN. We then explain how to create semantic data for TiAN.

3.1 Designing Timeline Action Network

To represent human activities in real-time, we add time information into TiAN. As shown in Figure 3, TiAN is expressed as a directed graph whose nodes are activity attributes, and whose edges are relations between these activity attributes. It is important to help the computers understand the meaning of data, thus we design TiAN based on OWL. Since N3 [11] is a compact and readable alternative to RDF’s XML syntax, we use N3 to describe data of TiAN.

To represent data of TiAN, we create classes and properties as below:

- ActionClass, ActClass, WhereClass, and WhatClass are classes of activity, act, location, object respectively.

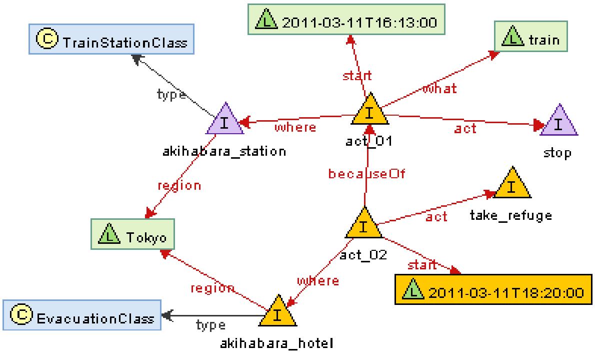


Fig. 3. An excerpt from timeline action network (C, I, L represent class, instance, label of activity attributes respectively)

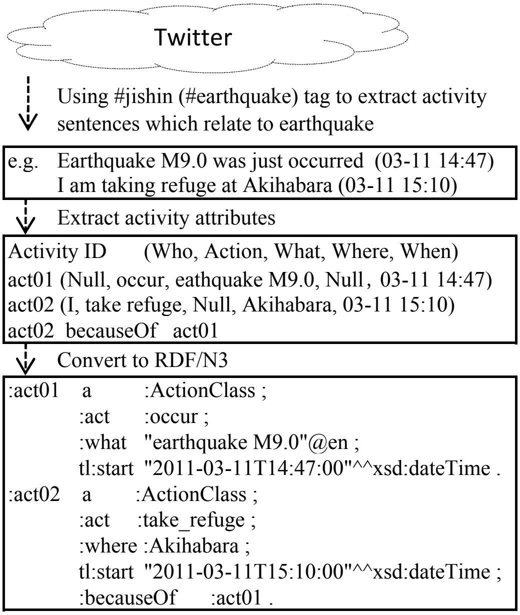
```
:stop                a                      :ActClass ;
                     rdfs:label             "stop"@en .

:akihabara_station   a                      :TrainStationClass ;
                     rdfs:label             "Akihabara station"@en ;
                     vcard:region           "Tokyo"@en ;
                     vcard:locality         "Chiyoda-ku"@en ;
                     vcard:street-address   "1-17-6 Sotokanda"@en ;
                     geo:lat                35.69858 ;
                     geo:long               139.773108 .

:act_01              a                      :ActionClass ;
                     :act                   :stop ;
                     :what                  "train"@en ;
                     :where                 :akihabara_station ;
                     tl:start   2011-03-11T16:13:00^^xsd:dateTime ;
                     tl:end     2011-03-11T23:45:00^^xsd:dateTime .
```

Fig. 4. An example of TiAN data

- EvacuationClass, ShopClass, RestaurantClass, TrainStationClass are classes of evacuation, shop, restaurant, train station respectively.
- TiAN has five properties: act, what, where, next, and becauseOf, which correspond to activity attributes, and relations between activities.

To easily link to external resource, TiAN inherits Geo [12], Time line [13], and vCards [14] ontologies. Geo [12] is used for representing latitude and longitude of a location. Time line [13] is used for representing time. And, vCards [14] is used for representing an address of a location.

Based on the above classes, properties, and inherited ontologies, we can describe data of TiAN. For example, Figure 4 represents the activity in the sentence “The train has stopped at Akihabara Station at 16:13:00”.


Twitter
Using #jishin (#earthquake) tag to extract activity sentences which relate to earthquake
e.g. Earthquake M9.0 was just occurred (03-11 14:47)
I am taking refuge at Akihabara (03-11 15:10)
Extract activity attributes
Activity ID (Who, Action, What, Where, When)
act01 (Null, occur, eathquake M9.0, Null, 03-11 14:47)
act02 (I, take refuge, Null, Akihabara, 03-11 15:10)
act02 becauseOf act01
Convert to RDF/N3

```
:act01   a         :ActionClass ;
         :act      :occur ;
         :what     "earthquake M9.0"@en ;
         tl:start  "2011-03-11T14:47:00"^^xsd:dateTime .
:act02   a        :ActionClass ;
         :act     :take_refuge ;
         :where :Akihabara ;
         tl:start  "2011-03-11T15:10:00"^^xsd:dateTime ;
         :becauseOf      :act01 .
```

Fig. 5. Method of creating semantic data for TiAN

3.2 Creating Semantic Data

Figure 5 explains how to create semantic data for TiAN. We first use #jishin (#earthquake) tag which relates to earthquake to collect activity sentences from Twitter. Secondly, we use our activity extraction method to extract activity attributes, and relationships between activities. Finally, we convert the extracted data to RDF/N3 to make semantic data for TiAN (Appendix 1).

4 Prediction of Missing Activity

Let $Can_{act} = \{act_1, act_2, .., act_t, ...\}$ is the set of candidate actions of the active user u_a at time t. Predicting the action of u_a at time t can be considered as choosing the action in Can_{act}, which has the most highest probability. Therefore, we need to calculate probability of u_a did act_t at time t ($P_{u_a \rightarrow act_t}$).

As shown in Figure 6, we can use collaborative filtering approach (CF) to calculate $P_{u_a \rightarrow act_t}$. However, while traditional CF [15,16,17] is trying to recommend suitable products on internet for users, our work tries to predict missing action data in real-world. Different from products, users' actions strongly depend on location, time, and before-after actions. Additionally, we need to consider execution time of each action. This means that it is not suitable to use traditional CF for our work.

Below, we propose a novel action-based CF to calculate $P_{u_a \rightarrow act_t}$.

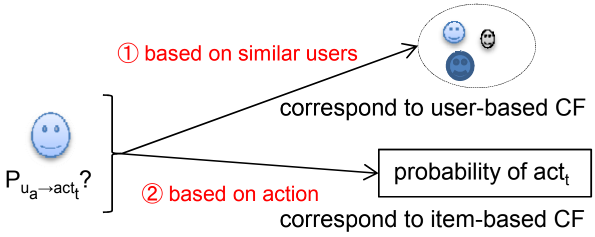


Fig. 6. Approach of predicting missing activity

4.1 Prediction Based on Similar Users

Based on the following ideas, we calculate similarity between two users in emergency situations.

- It is highly probable that as same as user u_a, similar users also did before action $(Did(a_{before}))$ and after action $(Did(a_{after}))$ of the candidate action act_tD
- If users had the same goal (e.g. wanted to evacuate in Shinjuku), then they had same action patterns $(SameTarget(a_t, l_t))$.
- It is highly probable that user did the same actions if they were in the same location $(SameLocation(l))$.

Therefore, the similarity between user u_j and user u_a will be calculated as Equation 1.

$$\begin{aligned} S(u_j, u_a) = &\beta Did\left(\{a_{before}, l_{before}\}, \{a_{after}, l_{after}\}\right) \\ &+ \gamma SameTarget\left(a_t, l_t\right) \\ &+ (1 - \beta - \gamma) SameLocation(l) \end{aligned} \tag{1}$$

Where:

- Parameters β, γ satisfy $0 \leq \beta, \gamma, \beta + \gamma \leq 1$. These parameters depend on each particular problem.
- If u_j did action act_t in location l, then $Did(act_t, l) = 1$, otherwise $Did(act_t, l) = 0$.
- If u_j and user u_a has the same goal (want to do action a_t in target location l_t), then $SameTarget(a_t, l_t) = 1$, otherwise $SameTarget(a_t, l_t) = 0$.
- If u_j and user u_a were in the same location l at the time t, then $SameLocation(l) = 1$, otherwise $SameLocation(l) = 0$.

4.2 Prediction Based on Probability of Action

In real-world, an action depends on location, time and its before-after actions. Therefore, probability of act_t at the time t can be calculated as Equation 2.

$$\begin{aligned} P(act_t) = &\rho_a \left\{F(a_{before} \to act_t) + F(act_t \to a_{after})\right\} \\ &+ \rho_t F(act_t, t) + (1 - \rho_a - \rho_t) F(act_t, l) \end{aligned} \tag{2}$$

Where:

- Parameters ρ_a, ρ_t satisfy $0 \leq \rho_a, \rho_t, \rho_a + \rho_t \leq 1$.
- $F(a_{before} \rightarrow act_t)$ is frequency of $a_{before} \rightarrow act_t$ (transition from a_{before} to act_t).
- $F(act_t \rightarrow a_{after})$ is frequency of $act_t \rightarrow a_{after}$ (transition from act_t to a_{after}).
- $F(act_t, t)$ is frequency of act_t at time t.
- $F(act_t, l)$ is frequency of act_t in location l.

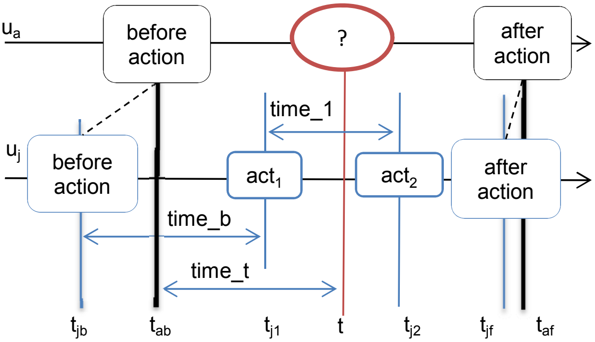


Fig. 7. An example of plural missing actions

4.3 Prediction Based on Execution Time of Action

Figure 7 explains an example of plural missing actions. In this example, we need to determine that u_a did act_1 or act_2 at time t. We can solve this problem based on the execution time of these actions done by user u_j.

Let $time\_b$, $time\_1$ are execution time of before action, act_1 done by the similar user u_j respectively. We suppose that it took the same execution time for u_a and u_j did the same action. Based on this supposition, we can use Equation 3 to determine that u_a did act_1 or not.

$$T(u_j, act_1) = \begin{cases} \frac{time\_t}{time\_b} & (\text{diff} < 0) \\ 1 & (0 \leq \text{diff} \leq \text{time\_1}) \\ \frac{time\_1}{diff} & (\text{time\_1} < \text{diff}) \end{cases} \tag{3}$$
$$where, diff = time\_t - time\_b.$$

4.4 Prediction Formula

Combination of Equation 1, Equation 2, and Equation 3, we can calculate $P_{u_a \rightarrow act_t}$ as Equation 4.

$$P_{u_a \rightarrow act_t} = \alpha \left(\frac{\sum_{j=1,L} \omega(u_j, act_t) * S(u_j, u_a) * T(u_j, act_t)}{L}) \right) \tag{4}$$
$$+ (1 - \alpha) P(act_t)$$

Where:

- L is number of all users similar to u_a.
- $\omega(u_j, act_t)$ is a weighting factor. If user u_j did act_t, then $\omega(u_j, act_t) = 1$, otherwise $\omega(u_j, act_t) = 0$.
- Parameters α satisfies $0 \leq \alpha \leq 1$. It depends on each particular problem.

4.5 Handling Minority Action

In emergency situations, successful actions are often minority actions, and have a great value. Therefore, we need a method that can handle not only the majority actions but also the minority actions. However, frequency-based method such as CF can easily handle the majority actions, but it is difficult to handle minority actions.

To deal with the minority action, we propose the following approach.

1. Using NLP to extract successful actions in feedbacks (tweets) from users. For example, if the users said that “it is a **good decision** when staying at company”, then we can consider “stay at company” as a successful action.
2. Predicting probability of u_a who did a successful action, based on the following idea.
   - Probability of u_a who did a successful action is proportional to percentage of successful actions.
   - The degree of success of an action is proportional to number of good feedbacks from the users.

   Therefore, we can calculate probability of u_a who did a successful action by Equation 5.

   $$DidSuccess_{u_a \to act_t} = f(u_a) * Success(act_t) \tag{5}$$

   Where:

   $$f(u_a) = \frac{number\ of\ successful\ actions}{number\ of\ actions}$$

   $$Success(act_t) = \frac{number\ of\ good\ feedbacks\ about\ act_t}{total\ number\ of\ good\ feedbacks}$$

Finally, the formula to predict missing action will be complemented as Equation 6.

$$P^*_{u_a \to act_t} = DidSuccess_{u_a \to act_t} + P_{u_a \to act_t} \tag{6}$$

5 Evaluation

In this section, we first evaluate our activity extraction approach. Secondly, we use SPARQL (SPARQL Protocol and RDF Query Language) [18] to evaluate our timeline action network. Then, we evaluate our proposed approach which predicts missing activities. Finally, we discuss the usefulness of the action network.

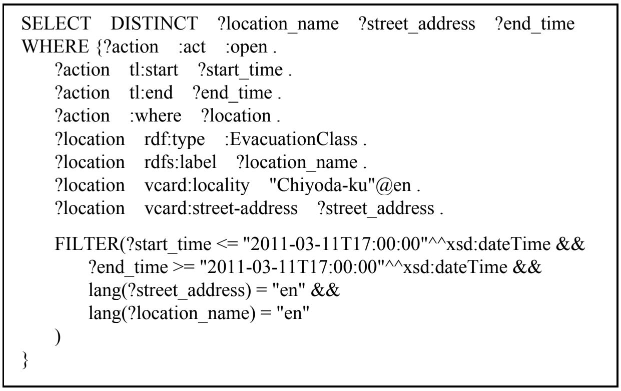
We collected 416,463 tweets which related to the massive Tohoku earthquake (Appendix 2). And then, to create data-set for the evaluations, we selected tweets which were posted by users in Tokyo from 2011/03/11 till 2011/03/12.

5.1 Activity Extraction

Table 1 shows the comparison results of our approach with baseline method, and Nguyen et al. [19]. Based on the results, we can see that the baseline has high precision but low recall. The reason is that sentences retrieved from twitter are often diversified, complex, syntactically wrong. Nguyen et al. also used self-supervised learning and CRF, but it could not handle complex sentences.

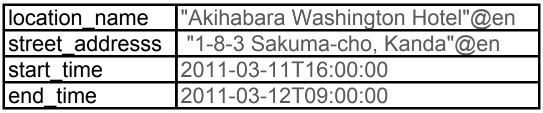
Table 1. Comparison of our approach with baseline, and Nguyen et al, (2011)

@	Method	Activity	Actor	Act	Object	Time	Location
Precision	Baseline	81.17%	86.32%	98.13%	84.14%	87.96%	88.25%
	Nguyen et al.	57.89%	72.79%	82.98%	67.01%	76.40%	80.20%
	Our approach	**73.21%**	**82.25%**	**97.11%**	**81.23%**	**80.04%**	**82.11%**
Recall	Baseline	23.86%	26.38%	28.87%	24.77%	26.20%	26.02%
	Nguyen et al.	51.13%	69.13%	90.23%	62.11%	73.51%	77.67%
	Our approach	**66.54%**	**80.11%**	**93.18%**	**76.57%**	**79.75%**	**81.02%**
F-measure	Baseline	36.88%	40.41%	44.61%	38.27%	40.37%	40.19%
	Nguyen et al.	54.30%	70.91%	86.45%	64.47%	74.93%	78.91%
	Our approach	**69.72%**	**81.17%**	**95.10%**	**78.83%**	**79.89%**	**81.56%**



```
SELECT   DISTINCT   ?location_name   ?street_address   ?end_time
WHERE {?action   :act   :open .
    ?action   tl:start   ?start_time .
    ?action   tl:end   ?end_time .
    ?action   :where   ?location .
    ?location   rdf:type   :EvacuationClass .
    ?location   rdfs:label   ?location_name .
    ?location   vcard:locality   "Chiyoda-ku"@en .
    ?location   vcard:street-address   ?street_address .

    FILTER(?start_time <= "2011-03-11T17:00:00"^^xsd:dateTime &&
        ?end_time >= "2011-03-11T17:00:00"^^xsd:dateTime &&
        lang(?street_address) = "en" &&
        lang(?location_name) = "en"
    )
}
```

Fig. 8. An example query for looking up opening evacuation centers based on time and location



location_name	"Akihabara Washington Hotel"@en
street_addresss	"1-8-3 Sakuma-cho, Kanda"@en
start_time	2011-03-11T16:00:00
end_time	2011-03-12T09:00:00

Fig. 9. An example result of opening evacuation center

5.2 Timeline Action Network

We used SPARQL to make RDF queries to evaluate our timeline action network. For example, Figure 8 shows the query that looks up an opening evacuation center based on the current time (2011-03-11T17:00:00), and the current location (Chiyoda-ku) of the active user. The result of this query is shown in Figure 9. Therefore, we can say that our action network is working properly with RDF queries.

5.3 Missing Activity Prediction

To evaluate our proposed approach, we first created correct action data of 3,900 Twitter users in Tokyo, after the massive earthquake occurred. Secondly, we repeated 10 times of the following experiment.

Table 2. Recall of Deleted Activity Data

Method	Action	Location	Action and Location
Baseline@	31.48%	43.09%	27.56%
Our method	**69.23%**	**76.92%**	**43.59%**

1. Randomly select 39 users as the active users.
2. Randomly delete activity data of these active users.
3. Consider the active users' names and time of deleted activities as input data, using our approach to determine whether the deleted activity data is reproduced or not.

The average results are shown in Table 2. In this table, baseline is the following method:

1. Look up the most similar user $most\_similar_{u_a}$ to the active user u_a.
2. Based on $most\_similar_{u_a}$, we predict missing activity of u_a.

From the above results, we can say that:

- Our approach can reproduce 69.23% of missing actions, 76.92% of missing locations, and 43.59% of missing activities (both of action and location).
- Not only the most similar user, our method considers all similar users and candidate actions. This is the reason why our method outperforms baseline.

5.4 Application of Timeline Action Network

If data on Twitter is real-time data, then we can say that TiAN reflects real-world activities in real-time. By using SPARQL (SPARQL Protocol and RDF Query Language), computers can grasp situations of trains, evacuation centers, food shops, etc. Therefore, we can use TiAN to find the nearest available evaluation center for disaster victims.

The computers also can recommend "what should to do" for the active user, based on action patterns of other users in TiAN. For example, as shown in Figure 10, based on past actions of other users, the computer can recommend {act_05, act_06, act_08} or {act_05, act_06, act_07, act_08} for the active user at Shinjuku station.

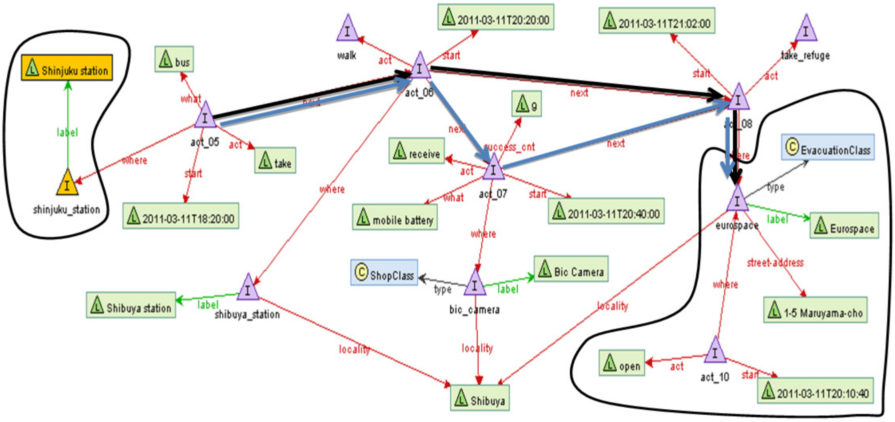


Fig. 10. An example of using TiAN to recommend action patterns for reaching to an available evacuation centers

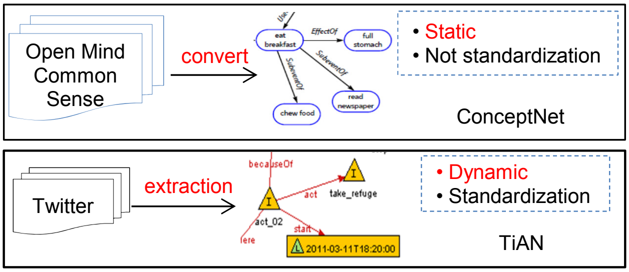


Fig. 11. Comparison with ConcepNet

6 Related Work

6.1 Action Network

ConcepNet (MIT Media Lab) [20] is a well known action network. This action network is a semantic network of commonsense knowledge, based on the information in OpenMind commonsense corpus (OMCS) [21].

Figure 11 shows comparisons of TiAN and ConcepNet. ConceptNet prepared a list of patterns in advance, and then it uses these patterns to obtain concepts, and the relations between these concepts. For example, given “A pen is made of plastic” as an input sentence, it uses “NP is made of NP” to get two concepts (a pen, plastic), and the relation (is made of) between these concepts. However, it is not practical to deploy this method for extraction of human activities from Twitter, because sentences retrieved from twitter are often diversified, complex, syntactically wrong. Additionally, TiAN is standardized based on OWL. Moreover, TiAN is a dynamic collaborative intelligence that represents instances of human activities in real-world.

6.2 Collaborative Filtering

[15,16] are the state-of-art approaches of the traditional CF. [15] proposed a combination item-based CF and user-based CF, but it did not consider time and location. [16] considered time, but did not consider location.

While traditional CF is trying to recommend suitable products on internet for users, our work is trying to predict missing action data in real-world. Different from products, user actions strongly depend on location, time, and before-after actions. Additionally, we need to consider execution time of each action. Moreover, we need to deal with minority actions.

7 Conclusions

In this paper, we have designed a timeline action network. Additionally, we used our previous work to automatically extract human activities from Twitter for this action network. We then proposed a novel action-based collaborative filtering, which predicts missing activity data. We also explained how to use this action network to assist disaster victims.

We are improving the approach of predicting missing activity data to complement the action network. We also need to consider reliability of activity data retrieved from Twitter as future work.

References

1. Matsuo, Y., Okazaki, N., Izumi, K., Nakamura, Y., Nishimura, T., Hasida, K.: Inferring long-term user properties based on users' location history. In: Proc. IJCAI 2007, pp. 2159–2165 (2007)
2. Poslad, S.: Ubiquitous Computing Smart Devices, Environments and Interactions. Wiley (2009) ISBN: 978-0-470-03560-3
3. Nguyen, T.M., Kawamura, T., Tahara, Y., Ohsuga, A.: Building a Time Series Action Network for Earthquake Disaster. In: Proc. ICAART 2012, vol. 1, pp. 100–108 (2012)
4. Nikkei Shimbun (2011), http://e.nikkei.com/e/fr/tnks/Nni20110705D05HH763.htm
5. Nakabayashi, I.: Development of urban disaster prevention systems in Japan - from the mid-1980s. Journal of Disaster Research 1 (2006)
6. Biglobe (2011), http://tr.twipple.jp/info/bunseki/20110427.html
7. Nguyen, T.M., Kawamura, T., Tahara, Y., Ohsuga, A.: Self-supervised capturing of users' activities from weblogs. International Journal of Intelligent Information and Database Systems 6, 61–76 (2012)
8. Banko, M., Etzioni, O.: The tradeoffs between open and traditional relation extraction. In: Proc. ACL 2008: HLT, pp. 28–36 (2008)
9. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proc. ICML, pp. 282–289 (2001)
10. Kudo, T., Matsumoto, Y.: Japanese dependency analysis using cascaded chunking. In: Proc. COLING, pp. 1–7 (2002)
11. W3C (2006), http://www.w3.org/DesignIssues/Notation3
12. Geo (2003), http://www.w3.org/2003/01/geo/wgs84_pos

13. Raimond, Y., Abdallah, S.: (2007), http://purl.org/NET/c4dm/timeline.owl
14. Halpin, H., Iannella, R., Suda, B., Walsh, N.: (2010), http://www.w3.org/2006/vcard/ns
15. Ma, H., King, I., Lyu, M.R.: Effective missing data prediction for collaborative filtering. In: Proc. SIGIR, pp. 39–46 (2007)
16. Koren, Y.: Collaborative filtering with temporal dynamics. In: Proc. KDD, pp. 447–456 (2009)
17. Sandholm, T., Ung, H.: Real-time, location-aware collaborative filtering of web content. In: Proc. CaRR, pp. 14–18 (2011)
18. W3C: SPARQL Query Language for RDF (2008)
19. Nguyen, T.-M., Kawamura, T., Nakagawa, H., Tahara, Y., Ohsuga, A.: Automatic extraction and evaluation of human activity using conditional random fields and self-supervised learning. Transactions of the Japanese Society for Artificial Intelligence 26, 166–178 (2011)
20. Liu, H., Singh, P.: Conceptnet: A practical commonsense reasoning tool-kit. BT Technology Journal 22, 211–226 (2004)
21. MIT (2011), http://openmind.media.mit.edu/

Appendix

1. An example of timeline action network: http://p.tl/q7MI

2. Tohoku earthquake data: http://p.tl/cw0W

The BOCHICA Framework for Model-Driven Agent-Oriented Software Engineering

Stefan Warwas[*]

German Research Center for Artificial Intelligence (DFKI),
Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany
stefan.warwas@dfki.de

Abstract. Modeling real world agent-based systems is a complex endeavour. An ideal domain specific agent modeling language would be tailored to a certain application domain (e.g. agents in virtual worlds) as well as to the target execution environment (e.g. a virtual reality platform). This includes the use of specialized concepts of the application domain, software languages (e.g. query languages for reasoning about an agent's knowledge), as well as custom views and diagrams for designing the system. This paper presents a model-driven framework for engineering multiagent systems, called BOCHICA. The framework is based on a platform independent modeling language which covers the core concepts of multiagent systems. In order to better close the gap between design and code, BOCHICA can be extended through several extension interfaces for custom application domains and execution environments. The framework is accompanied by an iterative adaptation process to gradually incorporate conceptual extensions. The approach has been evaluated at modeling agents in semantically-enriched virtual worlds.

1 Introduction

The research field of *Agent-oriented Software Engineering* (AOSE) is concerned with investigating how methods and algorithms developed in the wide area of *Artificial Intelligence* (AI) can be used for engineering intelligent software agents in a systematic way. AOSE should not be seen in isolation: As it gets increasingly applied in main stream software engineering it is confronted (of course) with typical problems of today's software development such as (i) an increasing number of software frameworks, programming languages, and execution platforms, (ii) shorter development cycles, and (iii) heterogeneous and distributed IT environments. A key to tackle the rapidly growing complexity in software development is abstraction. Higher-level software languages are required to hide the complexity and focus on the design of IT systems. *Model-driven Software Development* (MDSD) is driven by industry needs to deal with complex software systems. The underlying idea of MDSD is to model the *System Under Considerations* (SUC) on different levels of abstractions and use model transformations to gradually refine them until concrete code can be generated. Several core aspects of MDSD were standardized by the *Object Management Group* (OMG) as *Model-driven Architecture* (MDA).

[*] This paper was partially founded by the Saarbrücken Graduate School for Computer Science.

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 158–172, 2013.

During the recent years, several approaches for modeling agent-based systems have been proposed (e.g. [1]). Although the developed modeling languages are a step into the right direction, they have problems with fulfilling a user's need to efficiently model a custom application domain. An ideal modeling language would contain, beside the core concepts of *Multiagent Systems* (MAS), specific concepts, software languages, graphical representations, etc. for a certain target environment. The better the modeling language covers the target concepts, the less manual customizations are necessary at code level. Moreover, it is desirable to extend the modeling language with new concepts from AI and agent research (e.g. new ways of modeling interaction protocols). This paper proposes a model-driven framework for engineering agent-based systems which addresses the described problems. The framework, called BOCHICA[1], is based on a *Domain Specific Language* (DSL) which covers the main concepts of MAS. It provides several extension interfaces for integrating new concepts, AOSE methods, and 3rd party software languages. The customizations allow the user to tailor the framework to his needs without loosing the integration into a larger framework. The first part of this paper provides background on AOSE (Section 2), introduces the general idea of the BOCHICA framework (Section 3), and provides an overview of the extension interfaces (Section 4). The second part shows how to tailor the framework to the application domain of agents in semantically-enhanced virtual worlds and presents the evaluation results (Section 5). Finally, Section 6 discusses the related work and Section 7 concludes this paper.

2 Background

Raising the level of abstraction in software development was always an important driver in computer science research. The level of abstraction of a software language can be defined as the distance between the computer hardware and the concepts of that language [2]. Since the invention of computer systems, the level of abstraction was steadily increased from opcodes, assembler languages, procedural languages, up to object-oriented languages. The question that arises is how the next higher level of abstraction looks like. According to [2], "*The challenge for language engineers is that the software languages that we need to create must be at a higher level of abstraction, a level we are not yet familiar with. They must be well designed, and software developers must be able to intermix the use of multiple languages. Here too, we are facing a new, unknown, and uncertain task.*" In the agent community, AOSE has been seen as a natural successor of OOSE for a long time. Several articles discuss why AOSE has not arrived yet in mainstream software engineering [3][4][5]. Three of the identified main problems are (i) misunderstandings or wrong assumptions by non-agent experts, (ii) agent-oriented standards and methods are not yet sufficient for industry needs, (iii) lack of powerful tools. However, regarding the level of abstraction [3] concludes: "*With concepts such as roles and responsibilities, agent-oriented approaches to problem and system description are much closer to the ways in which managers conceive of business domains than are traditional software engineering descriptions.*" This matches the requirements stated by Kleppe. Our research hypothesis is

[1] Bochica was a semi-god of the Muisca culture who brought them living skills and showed them how to organize their lives.

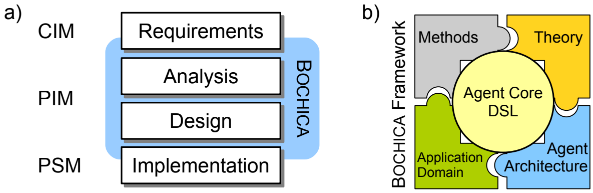


Fig. 1. a) depicts how BOCHICA relates to the abstraction layers defined by MDA (*Computational Independent Models (CIM), Platform Independent Models (PIM), Platform Specific Models (PSM)*). b) visualizes the idea behind the BOCHICA core DSL and 3rd party extensions.

that agent technology can embody concepts like goals, roles, and organizational structures in order to build modeling languages of the next higher level of abstraction.

3 Framework Overview

The role of BOCHICA in the overall software development process is to provide the means for capturing the design decisions of a SUC and bridging the gap between design and code. Figure 1 a) depicts how the framework is aligned to the abstraction layers defined by MDA. The agent-oriented modeling language underlying BOCHICA defines the concepts which are available for modeling a SUC (see Figure 1 b). So called *base transformations* are responsible for mapping the concepts of the BOCHICA core DSL to different agent platforms. In real world applications, an agent platform usually does not exist in isolation. As an agent platform is integrated into a larger execution environment, the core DSL gets extended with additional concepts to address the features of that execution environment. Moreover, a so called *extension transformation* defines additional conceptual mappings for the new concepts which complement an existing base transformation. The underlying idea is to reuse large parts of the existing infrastructure. The separation into a core modeling language and 3rd party extensions prevents the core DSL from getting cluttered with highly specialized concepts that are only relevant for a small number of applications (and thus, would make the language unusable over time). In the following, we provide a brief overview of the BOCHICA core DSL and introduce an iterative adaptation process for integrating conceptual extensions into BOCHICA.

3.1 Core DSL

The BOCHICA core DSL is based on the *Domain Specific Modeling Language for Multiagent Systems* (DSML4MAS) [6]. DSML4MAS is a platform independent graphical modeling language and covers the core aspects of MAS, such as agents and organizations, interaction protocols, goals, behaviors, deployment aspects, etc. Its abstract syntax is defined by the *Platform Independent Metamodel for Agents* (PIM4AGENTS). *Object Constraint Language*[2] (OCL)-based constraints are used for validating

[2] http://www.omg.org/spec/OCL/2.0/PDF/

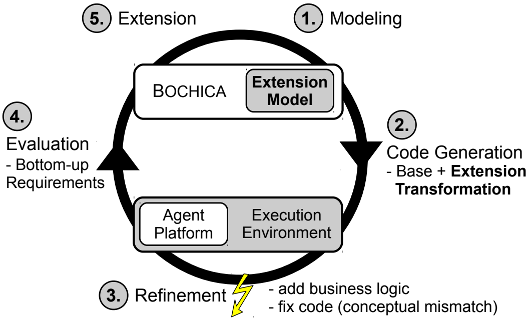


Fig. 2. This figure depicts the process for gradually adapting BOCHICA to a custom application domain

PIM4AGENTS models. Model validation on a platform independent level already prevents many errors in early phases of a project. DSML4MAS inherently possesses three different abstraction layers. The *macroscopic* layer defines the organizational structures of the SUC in terms of abstract goals, roles, interactions, and organizations. The *microscopic* layer defines agent types, behavior templates, concrete goals, etc. The *deployment* layer specifies concrete deployment configurations (e.g. agent instances and resources). The platform, domain, and methodology independent nature of DSML4MAS makes it the perfect language for building an extensible framework around it.

3.2 Iterative Adaptation

One of the main reasons in model-driven AOSE which causes design and code to diverge over time is that the modeling language is not expressive enough for capturing all design decisions. This makes extensive manual code refinements necessary at the platform level to address certain features of the target environment. To approach this problem, we propose an iterative adaptation process to gradually tailor BOCHICA to the needs of a custom application domain and execution environment (see Figure 2). In the first iteration, the SUC is modeled using the core DSL of BOCHICA without any conceptual extensions (step 1). An existing base transformation to a considered agent platform is applied (step 2). In step (3), the engineer realizes that some concepts of the execution environment cannot be addressed by the core DSL and/or the existing base transformation. The mismatches are analysed in step (4) and so called *bottom-up requirements* are collected. Finally, the new concepts and extension transformation are introduced in step (5). The second iteration makes use of the made extensions so that the gap between design and code is better closed. The adaptation process continues until the framework fits the needs of the SUC. It is important to note that the idea of the adaptation process is not to introduce arbitrary low-level concepts into BOCHICA. The extensions should complement the core DSL where necessary so that the design decisions can be captured. This task is not trivial and requires background on MDSD as well as on the BOCHICA core DSL.

4 Framework Interfaces

In order to customize the BOCHICA framework it offers various interfaces which can be extended through external Eclipse-based plug-ins. The remainder of this section provides an overview of those interfaces. Examples are provided by Section 5.

4.1 Conceptual Extension

BOCHICA can be extended with new concepts for (i) introducing new ways of modeling existing aspects (e.g. behaviors), (ii) introducing completely new aspects (e.g. commitments), or (iii) specializations for a certain application domain or execution environment. The extension is enabled by several interface concepts such as `Agent`, `Interaction`, `Resource`, and `Task` that can be specialized by external plug-ins. The benefit of extending our framework in opposite to creating a completely new approach is that large parts, which are common to most MAS, can be reused. The core concepts evolve over time and will build a solid foundation for AOSE. One example of how the BOCHICA framework can be extended is the approach for an alternative (declarative) way of modeling interaction protocols presented in [7]. The presented approach extended the `Interaction` concept and added custom diagrams. At the time of the creation of the extension, BOCHICA was not available so that DSML4MAS had to be extended directly. Now, BOCHICA provides interfaces for 3rd party developers for extending it with new concepts without touching the core. At the same time, the extension is integrated into the overall framework. End users can choose which alternative to apply. Technically, the extension mechanism is based on the Eclipse OSGi[3] framework and the *Eclipse Modeling Framework* (EMF) [8].

4.2 Data Model

Figure 3 depicts the data model interface of BOCHICA. The core of the data model has been separated from BOCHICA and is based on the Ecore metamodel provided by EMF [8]. Ecore is used to model classes and their attributes and relations among each other. The reuse of Ecore has several advantages: we get (i) graphical modeling support (UML class diagram style) and (ii) import from UML, XML schema (including XML de-/serialization) and existing Java code for free. Types defined in an Ecore-based data model can be made available within BOCHICA by the concept `EType`. On top of the Ecore metamodel, BOCHICA defines basic data structures such as `Sequence`, `Set`, or `HashMap`. Moreover, the data model interface also provides access to internal types such as `Event` and `Goal`. The internal types are required for accessing BOCHICA model artifacts inside a plan (e.g. the parameters of a goal). The data model can be extended by external plug-ins with specialized data structures. It is also possible to introduce an alternative to the Ecore metamodel for defining data types. Technically, the user defined data structures use the same extension interfaces as in Section 4.1.

[3] `http://eclipse.org/equinox/`

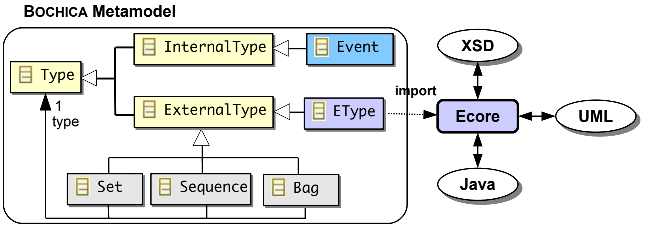


Fig. 3. The BOCHICA data model interface

4.3 Language Interfaces

There exists a large number of software languages that are relevant for developing agent-based systems such as (i) knowledge representation languages (e.g. OWL), (ii) query languages (e.g. SPARQL), (iii) rule languages (e.g. SWRL, PROLOG), (iv) communication languages (e.g. KIF, FIPA ACL), (v) programming languages (e.g. Java). A software language is always developed with a certain purpose in mind. Thus, it depends on the concrete use case which one to use. BOCHICA provides abstract language interfaces which can be extended by external language plug-ins (see Figure 4). The main concept is `Expression`. There exist several specialized expression types such as `BooleanExpression` and `ContextCondition`. The abstract expression types are used throughout the framework. For example, an `AchieveGoal` has a target and failure condition of type `BooleanExpression` and a `Plan` has a context condition of type `ContextCondition`. External plug-ins can specialize the abstract expression types with concrete languages. We assume that an external language is also based on Ecore. This is not a hard restriction since more and more software languages, such as SPARQL or Java, are becoming available in public metamodel zoos (e.g. EMFText concrete syntax zoo[4], Atlantic metamodel zoo[5]). We use a reflection-based approach for parsing user defined expression strings into a language specific expression model (interface concept `EObject`) and assign it to the `Expression` object's `object` attribute (see Figure 4). This approach can be used (i) for checking the syntactical correctness of an expression, (ii) for checking whether variable symbols inside the language model are bound in the surrounding scope, and (iii) to process the expression models in model transformations. The benefit of our approach is that technical details, such as the integration of the knowledge base and SPARQL into the concrete agent execution platform, are hidden on the modeling level. At the same time, models can be tailored to a certain target environment. Of course, the integration at the platform level has to be done at some point (we discuss it later) but the agent engineer has an abstract view and can concentrate on the design of the overall system.

4.4 Methodologies

During the recent years, several agent-oriented methodologies have been proposed [9]. Most of the developed approaches are supported by a graphical modeling language

[4] `http://www.emftext.org/`

[5] `http://www.emn.fr/z-info/atlanmod/index.php/Atlantic`

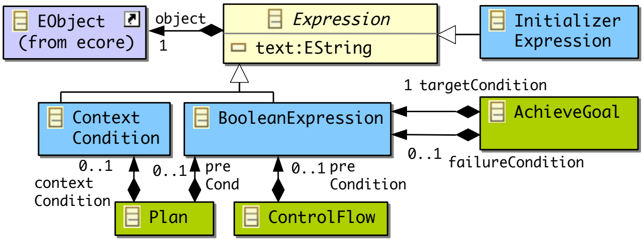


Fig. 4. Expression interface of BOCHICA

(see discussion in Section 6). The focus of our approach was always on developing an expressive platform independent agent modeling language that can be used for model-driven development of agent-based systems and less on the methodology part. Since both aspects are complementary, BOCHICA can be extended by AOSE method plug-ins to support the software development process. In the same way as BOCHICA can be extended with new agent concepts, methodology providers can contribute plug-ins with new views and methodology concepts. For example, the Prometheus methodology [10] uses so called *scenarios* in the *system specification* phase to identify typical events, actions, and goals of the SUC. In the *architectural design* phase, the collected entities are grouped to roles and agents. A Prometheus plug-in for BOCHICA could extend it with the missing concepts for modeling scenarios in the system specification phase (since it is not covered by the core DSL). The architectural design phase could be based on existing concepts of the core DSL. As interface, BOCHICA provides the concept `MethodArtifact`. Instead of having a separate modeling language and tool for each methodology, most of the methodologies could be integrated into one framework and share a common core. This would join the efforts of the involved parties and would ease the maintenance of the tool chain.

4.5 Transformations

Model transformations in MDSD are used to gradually refine a model of a SUC until executable code is generated. We assume that there exist *base transformations* from BOCHICA to agent execution platforms such as Jack or Jadex. As BOCHICA gets extended with new concepts, an existing base transformation is no longer complete regarding the covered concepts. Thus, an *extension transformation* is required which extends an existing base transformation for the new concepts (if the target platform shall be enabled for the extension). We see three possibilities how this can be achieved. Some model transformation languages (e.g. QVT[6]) allow to write a new transformation which inherits from an existing one. Thus, an existing mapping rule can be overloaded by a new and extended one. Other transformation languages like XPand[7] use an aspect-orientated approach for hooking into an existing transformation and extending it. A further possibility is to chain transformations, where the first one is a base

[6] `http://www.omg.org/spec/QVT/1.0/`

[7] `http://www.eclipse.org/modeling/m2t/`

transformation and the succeeding one supplements the result of the proceeding one. Thus, an external plug-in for BOCHICA usually consists of (i) conceptual extensions and (ii) an extension transformation for the required target environment (assuming that the base transformation already exists). Maintaining the tool chain is one of the main problems in MDSD. Reusing existing model transformations reduces development costs and time and increases code quality by using well established design patterns.

4.6 Custom Views and Tools

Views are used in graphical modeling languages to visualize the relations of model artifacts of a certain sub-aspect of a system. BOCHICA provides standard views for agent types and organizational structures, protocols, goal hierarchies, deployment configurations, etc. 3rd party developers can use the extension interface for customizing BOCHICA to a certain application domain or introduce new ways of viewing existing aspects. Views can also help to adapt the development environment to certain user groups. Technically, diagrams and tools can be plugged into the framework by using the extension point mechanism of the Eclipse OSGi framework and GMF[8].

5 Evaluation

This section evaluates the BOCHICA framework in a complex real world case study. As of today, intelligent behavior of avatars in virtual worlds is usually simulated by triggered script sequences which create the illusion of intelligent behavior for the user. However, the flexibility of those avatars is, due to their static nature, very limited. In the research project *Intelligent Simulated Realities* (ISReal) our research group developed a simulation platform based on semantic web technology for bringing intelligent behavior into virtual worlds [11]. The basic idea of ISReal was to use semantic web technology to extend purely geometric objects with ontological information (OWL-based; e.g. concept door links two rooms and can be open or closed) and specify their functionality by semantic service descriptions (OWL-S-based; e.g. open and close door services), called *object services*. Intelligent agents perceive this information, store it in their knowledge base, and use it for reasoning and planning. An object service is grounded in a service implementation which invokes according animations or simulation modules. The platform consists of various simulation components which can be distributed in a network. Before we discuss the BOCHICA extensions for developing intelligent ISReal avatars, we introduce the main components of the ISReal platform.

Global Semantic Environment. The *Global Semantic Environment* (GSE) maintains the global ontological facts of the virtual world. It is responsible for (i) executing object services (e.g. checking the pre-condition and invoking the service grounding), (ii) updating facts (e.g. when a door gets closed), and (iii) handling queries (e.g. SPARQL).

Agent Environment. The ISReal agent environment defines interfaces for connecting 3rd party agent execution platforms to the ISReal platform (we currently use Jack,

[8] http://www.eclipse.org/gmf

Jadex, and the Xaitment[9] game AI engine). Every ISReal agent is equipped with a *Local Semantic Environment* (LSE) which is an agent's local knowledge base. The LSE stores the perceived information and enables the agent to reason about it. Moreover, the LSE is equipped with an AI planner.

Graphics Environment. The user interface of the ISReal platform is realized by a XML3D[10]-enabled standard web browser. The 3D scene graph is part of the browser's *Document Object Model* (DOM) and can be manipulated using Java Script. It also contains RDFa[11]-based semantic annotations of the 3D-objects such as the concept URI, the object URI, and the semantic object service URIs. Moreover, we extended the browser with an agent sensor which allows agents to perceive the annotated 3D objects.

An intelligent ISReal avatar consists of (i) the geometrical shape (body) and animations, (ii) a perception component, (iii) semantic annotations, and (iv) an agent that processes the perceived information and controls the body. Artifacts such as the geometrical shape, animations, or ontologies are developed using specialized 3rd party tools. We decided to base the development environment for ISReal agents on BOCHICA and use Jadex as the target agent platform. This has several advantages: (i) BOCHICA already provides the core concepts, diagrams, etc. for modeling agent systems, (ii) we can reuse the existing base transformation to Jadex, (iii) we only need to customize the missing aspects of BOCHICA for creating an individual development environment for agents in semantic virtual worlds, and (iv) it enables the reuse of existing model artifacts (e.g. interaction protocols). Figure 5 depicts the big picture of how we think that intelligent agents for the ISReal platform should be developed. For a detailed introduction to the ISReal platform we refer to [11]. The remainder of this section discusses the extensions of the BOCHICA framework for developing ISReal agents.

5.1 Conceptual Extension

Figure 6 depicts some of the conceptual extensions for ISReal. The upper row shows interface concepts of BOCHICA. The `OMSConfig` concept is the root of a metamodel which is used in the ISReal platform for configuring the LSE with concrete ontologies, object services, etc. The imported `OMSConfig` concept of the ISReal platform is reused by the extension plug-in. The bottom row depicts the actual conceptual extensions. The `ISRealAgent` specializes the concept `Agent` and has an URI which defines an agent's ontological type, an `ISRealRaySensor` (resolution, update rate), a LSE, and a (not visualized) reference to an existing graphical avatar (the agent's body). Some concepts of BOCHICA change their technical meaning when they are transformed to the ISReal platform. For example, ISReal agents use their plans to orchestrate object services. BOCHICA already provides support for orchestrating web services by plans. Since ISReal object services are very similar to web services, the existing concepts can be reused without modification. Figure 7 depicts a simple plan that is triggered by the `MoveNearGoal` and invokes the `MoveNearService` with the according parameters. The `MoveNearService` is used for in-room navigation (no path finding across

[9] `http://www.xaitment.com/`

[10] `http://www.xml3d.org`

[11] `http://www.w3.org/TR/xhtml-rdfa-primer/`

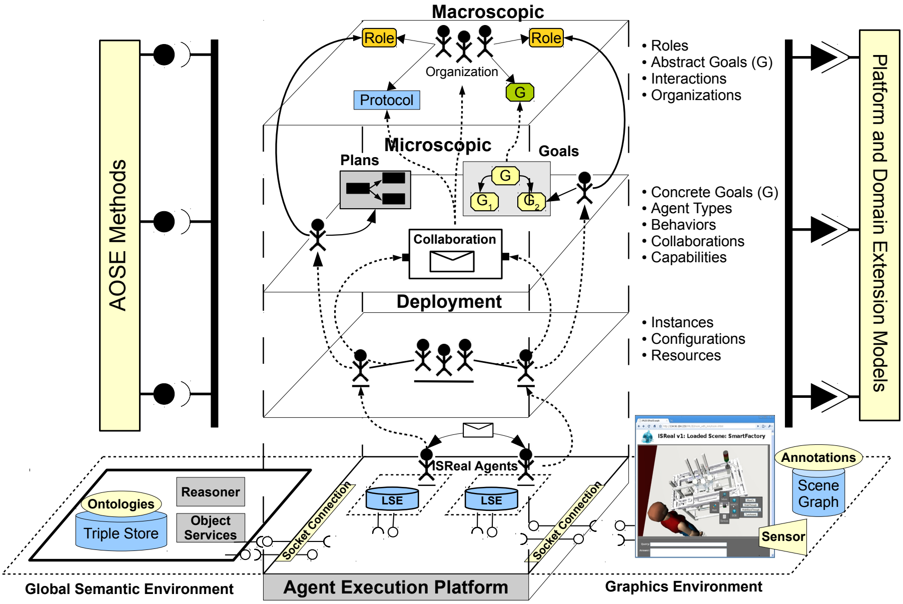


Fig. 5. The bottom layer depicts the components of the ISReal platform. The upper central part shows the inherent degrees of abstraction of BOCHICA. The left an right hand side represent the interfaces for extending the framework.

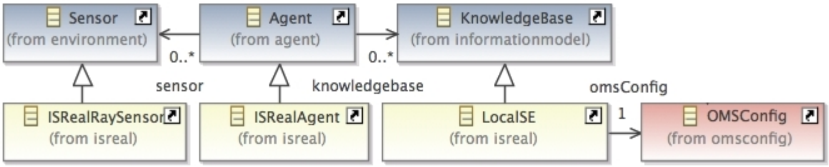
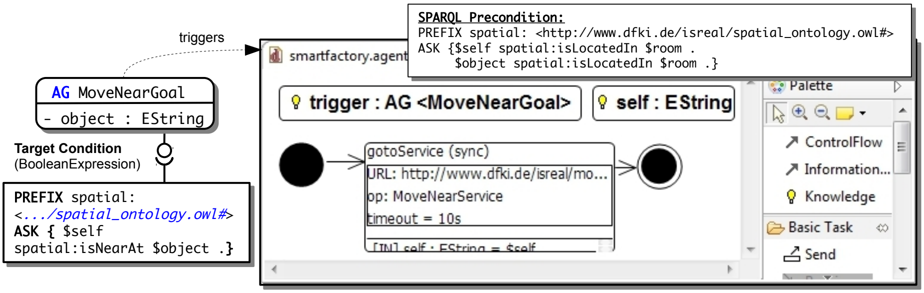


Fig. 6. A part of the ISReal extension of BOCHICA



```
SPARQL Precondition:
PREFIX spatial: <http://www.dfki.de/isreal/spatial_ontology.owl#>
ASK {$self spatial:isLocatedIn $room .
     $object spatial:isLocatedIn $room .}
```

```
PREFIX spatial:
<.../spatial_ontology.owl#>
ASK { $self
spatial:isNearAt $object .}
```

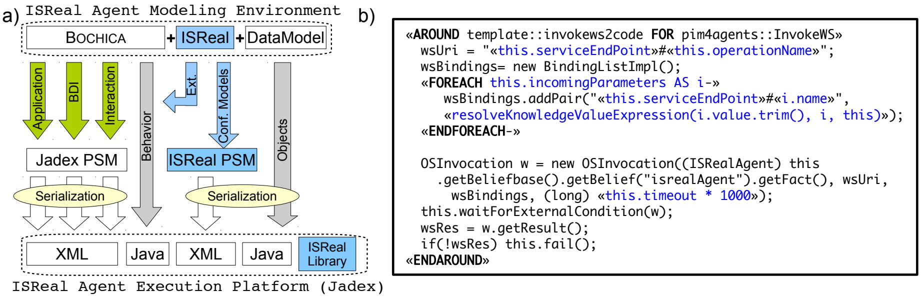
```
gotoService (sync)
URL: http://www.dfki.de/isreal/mo...
op: MoveNearService
timeout = 10s
[IN] self : EString = $self
```

Fig. 7. Agent-based orchestration of ISReal object services (behavior diagram)

rooms). The plan's precondition checks whether the target object is located in the same room. The ISReal model transformation generates code for invoking an ISReal object service instead of code for invoking an ordinary web service (see Section 5.3).

5.2 Language Extension

In order to make rational decisions, it is essential for agents to reason about the perceived information. The interface to a knowledge base is usually defined by a query language. As ISReal agents are based on semantic web technology, we decided to use SPARQL queries to access the LSE. Two of the application scenarios are (i) to use SPARQL-Ask queries to define the target condition of achieve goals and (ii) to specify the context condition of plans with SPARQL-Select queries. As explained in Section 4.3, BOCHICA defines language interface concepts such as `BooleanExpression` that are used throughout the framework. We reused an Ecore-based SPARQL DSL which is provided by the EMFText concrete syntax zoo to extend BOCHICA (see Section 4.3). The `BooleanExpression` was extended with SPARQL-Ask and the `ContextCondition` with SPARQL-Select. We also reused the automatically generated parser of EMFText for parsing SPARQL text queries into SPARQL models that are plugged into the BOCHICA extension slot. Figure 7 depicts an example `AchieveGoal` for walking to an object. The target condition `isNearAt(self, object)` is validated in the agent's LSE. The predicate is perceived by the agent through its sensor after it has been computed by the graphics environment and the GSE.



```
«AROUND template::invokews2code FOR pim4agents::InvokeWS»
  wsUri = "«this.serviceEndPoint»#«this.operationName»";
  wsBindings= new BindingListImpl();
  «FOREACH this.incomingParameters AS i-»
     wsBindings.addPair("«this.serviceEndPoint»#«i.name»",
     «resolveKnowledgeValueExpression(i.value.trim(), i, this)»);
  «ENDFOREACH-»

  OSInvocation w = new OSInvocation((ISRealAgent) this
    .getBeliefbase().getBelief("isrealAgent").getFact(), wsUri,
     wsBindings, (long) «this.timeout * 1000»);
  this.waitForExternalCondition(w);
  wsRes = w.getResult();
  if(!wsRes) this.fail();
«ENDAROUND»
```

Fig. 8. a) depicts an overview of the transformation architecture and b) an example mapping rule in XPand

5.3 Transformations

A Jadex application consists of XML-based configuration files for applications, agents, and capabilities. Behaviors are encoded in Java-based plans. The base transformation from BOCHICA to Jadex consists of the four modules Application, BDI, Interaction, and Behavior (see Figure 8 a). The first three modules map concepts from BOCHICA to the Jadex *Platform Specific Metamodel* (PSM) (green arrows) using QVT model-to-model transformations. The generated Jadex model is automatically serialized to valid Jadex XML files by EMF. We decided not to create a separate Jadex metamodel for plans. This decision was made due to experiences with previous transformations to Jack and

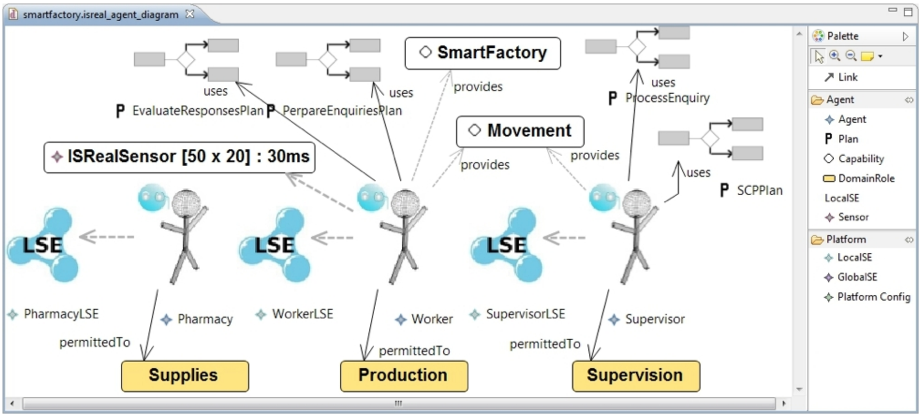
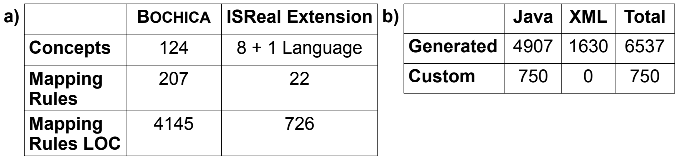


Fig. 9. The customized ISReal agent diagram

Jade (to avoid overhead and simplify extensions). The model-to-text transformation is done using XPand. The separation of the transformation into modules leverages extensibility and eases maintenance. As explained in Section 4.5, the data model is based on Ecore and we rely on the capability of EMF to automatically serialize types to Java code (white arrow). Since we want to focus on the overall approach, the details of the transformations and the Jadex PSM will not be discussed in this paper. The blue parts in Figure 8 depict (i) the ISReal extension of BOCHICA, (ii) the ISReal extension to the Jadex QVT and XPand base transformations, (iii) the generation of configuration files for the ISReal agent component, and (iv) an additional ISReal library that enables Jadex for the ISReal platform. The ISReal library implements the interfaces of the ISReal agent environment for passing incoming perception events and user queries to the agents running in the Jadex platform. Moreover, it includes Jadex into the start-up procedure of the distributed ISReal platform and provides an ISReal capability which makes a Jadex agent to an ISReal agent. For example, it equips an agent with a LSE. Figure 8 b) depicts a XPand-based aspect-oriented mapping rule which replaces the original mapping rule of the Jadex base transformation for invoking a standard web service by the invocation of an ISReal object service. The first part sets the variable bindings of the object service and the second part does the actual invocation through a helper class provided by the ISReal library.

5.4 ISReal View

The technical details explained so far are (in the ideal case) not visible to the end user. He is guided by a methodology and uses graphical diagrams to design a MAS for a certain use case. The graphical front end abstracts from technical details such as (i) the integration of Jadex into the ISReal platform, (ii) the invocation of ISReal object services in Jadex, or (iii) the evaluation of SPARQL queries in the LSE. Figure 9 depicts the ISReal agent diagram which contains, in addition to the standard BOCHICA agent diagram artifacts, the ISReal sensor and the LSE. Placing an ISReal agent implies, compared to a plain agent, the generation of an ISReal agent component which integrates



a)

	BOCHICA	ISReal Extension
Concepts	124	8 + 1 Language
Mapping Rules	207	22
Mapping Rules LOC	4145	726

b)

	Java	XML	Total
Generated	4907	1630	6537
Custom	750	0	750

Fig. 10. Table a) compares the BOCHICA core DSL and the Jadex base transformation with the ISReal-specific extensions. b) compares the generated code for the SmartFactory case study to the manually written *Lines of Code* (LOC).

into the ISReal platform, the LSE, sensor interfaces, different service execution semantics, etc.

The ISReal-enabled BOCHICA framework has been evaluated in a virtual production line scenario. For this purpose, a virtual representation of the DFKI SmartFactory[12] was semantically annotated. The SmartFactory is a real existing machine of the DFKI to evaluate technology for the factory of the future. The BOCHICA ISReal extension has been used to model agents that perform typical workflows such as handling new orders and fixing problems as they occur. Figure 9 depicts an overview of the modeled ISReal agents. The current model encompasses basic navigation and object interaction (e.g. for operating the machine). In order to estimate the effort for the ISReal customization, Figure 10 depicts two tables. Table a) compares (i) the number of concepts, (ii) the number of transformation rules, and (iii) the *Lines of Code* (LOC) of the model transformation between BOCHICA and the ISReal extension. Table b) compares the generated code with the manually written code for the SmartFactory scenario. The manual code mainly implements business logic (e.g. computing the shortest path between two objects). The ISReal extension required around 10% new concepts and mapping rules. The result is a modeling environment which addresses the needs of one specific application domain. BOCHICA is especially suited for large and medium size application domains and target environments with many end-users (e.g. the ISReal platform) where customizations pay off. Small applications can be realized with the functionality provided by the core modeling language and the base transformations (similar to existing approaches). Probably the biggest obstacle of the approach is the required additional knowledge of the BOCHICA core DSL and MDSD.

6 Related Work

During the recent years, several agent-oriented modeling languages have been proposed. The majority of the modeling languages were created in order to support a certain agent methodology [1] [9]. One problem of existing methodology-oriented modeling approaches that we see is that they do not clearly distinguish between (i) the agent platform, (ii) the methodology, and (iii) the modeling language. Two indicators which support our perception are (i) the development of the modeling languages is not decoupled from the methodologies and (ii) none of the languages has an own name (only the tools have names). However, we think the development of modeling languages is

[12] `http://smartfactory.dfki.uni-kl.de/de/`

orthogonal to the development of agent methodologies and tools. Of course, a methodology can (and most likely will) have certain requirements to a modeling language (e.g. own methodology artifacts and views). For this purpose, BOCHICA can be extended with methodology artifacts. However, the core of the agent modeling language is independent of a certain methodology. Unfortunately, the majority of the developed modeling tools are only partially based on standardized technology for model-driven development which hampers the benefits of MDSD. For example, the *Prometheus Design Tool*[13] (Prometheus methodology) has no explicit underlying metamodel. Others like AgentTool III[14] (O-MaSE), INGENIAS Development Kit[15] (INGENIAS), Taom4e[16] (Tropos) are only partially based on MDSD (e.g. proprietary or non-MDA-based model transformations). To the best of our knowledge, the mentioned approaches do not consider extensibility as presented by this paper. Beside the methodology-based modeling languages, there exist also approaches for extending the *Unified Modeling Language* (UML) with agent concepts (e.g. *Object Management Group's* (OMG) *Agent Metamodel and Profile*[17] (AMP) or FIPA Agent UML[18]). Those approaches promise to reuse the ecosystem built around UML – including the large user group. However, modeling agents is fundamentally different from modeling objects. Agents possess an internal cognitive model and require different methods and design patterns. Moreover, our experiences in AMP showed that it is hard to extend UML since the underlying *Meta Object Facility* (MOF) metamodel is complex and extensions of existing elements have many not desired and non-obvious implications. Thus, we are sceptical that extending UML in its current form suffices the needs of AOSE. UML, which is a general purpose modeling language, offers two extension mechanisms: (i) heavy weight metamodel extensions and (ii) light weight profiles. Metamodel extensions of UML underlie the standardization process of OMG and are not for the normal end user. Profile-based extensions can be created by end users and allow a limited customization. An alternative to our approach would be the creation of a platform specific modeling language (e.g. for the ISReal-enabled Jadex platform). This would mean to reinvent many things that are already part of BOCHICA. In [12] two platform specific modeling languages for the agent platforms SEAGENT [13] and Jadex were presented. The possibility to customize the language if the agent platform (e.g. Jadex) is integrated into a larger platform is not discussed.

7 Conclusions

In this paper, we presented a novel model-driven framework for AOSE which integrates the experiences we gained during the recent years with modeling MAS. The BOCHICA framework goes beyond the state of the art in AOSE as it is not created for a certain execution platform, methodology, or application domain. Instead, it is based

[13] `http://www.cs.rmit.edu.au/agents/pdt/`

[14] `http://agenttool.cis.ksu.edu/`

[15] `http://ingenias.sourceforge.net/`

[16] `http://selab.fbk.eu/taom/`

[17] `http://www.omg.org/cgi-bin/doc?ad/08-09-05.pdf`

[18] `http://www.auml.org/`

on a platform independent agent core DSL and provides generic extension interfaces for integrating results from agent research as well as for customizing it regarding user-specific application domains, AOSE methods, and target platforms. Section 3 provided an overview of BOCHICA and presented an iterative adaptation process for integrating conceptual extensions. The framework interfaces were introduced in Section 4. We evaluated BOCHICA at the application domain of agents in semantically-enhanced virtual worlds. Our experience showed that around 10% of new concepts and mapping rules were necessary to create a development environment which enables efficient modeling of ISReal agents. We see our approach as a contribution to the unification of the diverse field of agent-oriented modeling and to bridge agent research and concrete software development. In the future, we want to integrate existing agent methodologies and work on collaborative modeling of agent-based systems.

References

1. Henderson-Sellers, B., et al.: Agent-Oriented Methodologies. IGI Global (2005)
2. Kleppe, A.: Software Language Engineering: Creating Domain-Specific Languages Using Metamodels. Addison-Wesley Longman, Amsterdam (2008)
3. Belecheanu, R.A., et al.: Commercial applications of agents: Lessons, experiences and challenges. In: 5th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems, pp. 1549–1555 (2006)
4. Jennings, N.R., Wooldridge, M.: Agent-Oriented Software Engineering. Artificial Intelligence 117, 277–296 (2000)
5. McKean, J., et al.: Technology diffusion: analysing the diffusion of agent technologies. Autonomous Agents and Multi-Agent Systems 17, 372–396 (2008)
6. Hahn, C., et al.: A platform-independent metamodel for multiagent systems. Autonomous Agents and Multi-Agent Systems 18, 239–266 (2009)
7. León-Soto, E.: A Model-Driven Approach for Executing Modular Interaction Protocols Using BDI-Agents. In: Fischer, K., Müller, J.P., Levy, R. (eds.) ATOP 2009 and ATOP 2010. LNBIP, vol. 98, pp. 10–34. Springer, Heidelberg (2012)
8. Steinberg, D., et al.: EMF: Eclipse Modeling Framework, 2nd revised edn. Addison-Wesley (2008)
9. Sterling, L., et al.: The Art of Agent-Oriented Modeling. The MIT Press (2009)
10. Padgham, L., et al.: Developing Intelligent Agent Systems: A Practical Guide. John Wiley & Sons (2004)
11. Kapahnke, P., Liedtke, P., Nesbigall, S., Warwas, S., Klusch, M.: ISReal: An Open Platform for Semantic-Based 3D Simulations in the 3D Internet. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part II. LNCS, vol. 6497, pp. 161–176. Springer, Heidelberg (2010)
12. Kardas, G., Ekinci, E.E., Afsar, B., Dikenelli, O., Topaloglu, N.Y.: Modeling Tools for Platform Specific Design of Multi-Agent Systems. In: Braubach, L., van der Hoek, W., Petta, P., Pokahr, A. (eds.) MATES 2009. LNCS, vol. 5774, pp. 202–207. Springer, Heidelberg (2009)
13. Dikenelli, O.: SEAGENT MAS platform development environment. In: Proc. of the 7th Int. Joint Conf. on Autonomous Agents and Multiagent Systems: Demo Papers, AAMAS 2008, pp. 1671–1672. IFAAMAS (2008)

Combining Uniform and Heuristic Search: Solving DSSSP with Restricted Knowledge of Graph Topology

Sandro Castronovo[1], Björn Kunz[1], and Christian Müller[1,2]

[1] German Research Center for Artificial Intelligence (DFKI),
Campus D3.2, Saarbrücken, Germany
[2] Action Line Intelligent Transportation Systems, EIT ICT Labs, Germany

Abstract. Shortest-path problems on graphs have been studied in depth in Artificial Intelligence and Computer Science. Search on dynamic graphs, i.e. graphs that can change their layout while searching, receives plenty of attention today – mostly in the planning domain. Approaches often assume global knowledge on the dynamic graph, i.e. that topology and dynamic operations are known to the algorithm. There exist use-cases however, where this assumption cannot be made. In vehicular ad-hoc networks, for example, a vehicle is only able to recognize the topology of the graph within wireless network transmission range. In this paper, we propose a combined uniform and heuristic search algorithm, which maintains shortest paths in highly dynamic graphs under the premise that graph operations are not globally known.

1 Introduction

Shortest path problems on graphs have been thoroughly studied in Artificial Intelligence and Computer Science literature. Single-source shortest path or all-pair shortest path problems with positive edge weights can be solved using widely known algorithms, such as Dijkstra [1] and A*. There are a large number of applications: modern navigation systems, for example, use implementations of these algorithms for route planning, network protocols use them in order to route data packets from one physical location to another and many planning systems in Artificial Intelligence use variants of these algorithms.

The problem becomes significantly more challenging when we allow certain dynamic operations on the graph while searching, such as insertions or deletions of vertices as well as changes of edge weights. This problem drew attention of research in Artificial Intelligence and related Computer Science fields. Approaches by [2], [3], [4], [5], [6], [7] among others assume global knowledge on the performed graph operations. This means they are only applicable on graphs, for which the topology and all executed operations are known at every point in time. [3], for example, assume that every vertex stores its distance to a start vertex of a shortest path. They modify this information on every graph operation and use the result for re-calculating. Work by [4] transfers the problem into the domain of Learning Automata but also allows access to the entire topology of the dynamic graph.

There exist applications, however, in which this knowledge can neither be assumed nor derived. Consider a vehicular ad-hoc network, for example, where vertices are vehicles, edges are wireless connections between two vehicles, the task is to send data

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 173–187, 2013.

from car A to car B, and B is not in A's transmission range. While searching a path to B, all of the operations above can occur in arbitrary order and number, because the topology of the underlying graph is subject to fast deletion and insertions of vertices and the algorithm has no information about these changes. Hence, we look at a "fully dynamic problem" (Dynamic Single Source Shortest Path Problem, or DSSSP). In our example, the dynamic operations may even break the connection between vertices. Furthermore, it is not efficient and sometimes even not feasible to visit every vertex after a graph operation. Network bandwidth is considered a limited resource in vehicular ad-hoc networks, which is also a reason why the cited algorithms are not applicable here.

In this paper, we propose a combined uniform and heuristic search algorithm, which is able to solve the single-source shortest path problem under *unknown* topology changes in fully dynamic graphs. In order to do so we instantiate two graphs: The first, $G = (V, E)$, is static. We solve the single-source shortest path problem on G using Dijkstra's algorithm. The second, $\tilde{G} = (\tilde{V}, \tilde{E})$, is subject to the described graph operations. We then exploit *domain specific* relations between the two graphs and approximate the shortest path identified on G in $\tilde{G}$. Our algorithm only depends on *local* information about graph operations in order to heuristically maintain or alter the initial path found in G.

2 Related Work

[3] propose an incremental version of the popular A* algorithm based on a dynamic gridworld. This world consists of cells which denote vertices. Edges are drawn between neighboring cells. The algorithm continuously finds shortest paths between a start cell s_{start} and a goal cell s_{goal}. Their approach is incremental since they reuse results from previous searches. However, they assume global knowledge about the gridworld: The algorithm holds data structures for the start distance of every cell as well as a list of traversable and blocked cells. In our application domain this assumption cannot be made.

The approach of [4] transfers the DSSSP problem into the domain of Learning Automata. They establish three components: The learning automaton (LA), which is instantiated in every vertex, the random environment (RE) and a penalty/reward system. RE changes edge weights using stochastic information and LA constantly interacts with this environment by guessing whether or not a node belongs to the shortest path. The LA constantly receives rewards or penalties depending on whether its guess was right or not. Although the distribution of the weight changes by the RE are unknown to LA, it is allowed to retrieve a snapshot of the whole graph and its current edge weights. Basically, this snapshot, Dijkstra's algorithm and the received penalties/rewards are used for shortest path computation. It is obvious that in our domain such a snapshot of the whole graph is not available.

[7] propose fully dynamic algorithms for solving DSSSP by counting the vertices affected by changes of the graph. When increasing the weight of an edge, the affected vertices are those which change the distance from the start vertex. The algorithm marks vertices according to their status after a graph operation; White marked vertices were

not affected by an operation. Neither the distance from the source nor the parent in the shortest path tree changed. Red marked vertices increased their distance from the source while the distance of the ones marked pink remained constant but it replaces the old parent in the shortest path tree. Obviously, this approach also relies on global knowledge about the graph topology and the operations on it.

[8] (SEET) and [9] (GRANELLI) are both routing protocols for vehicular ad-hoc networks. Both implementations have to solve the single-source shortest path problem in dynamic graphs having no knowledge about its topology by the very nature of their application domain. The approach by [8] is to employ two graphs: One fixed, one dynamic. There exists a connection between the two in a way such that the topology of the fixed graph correlates with the dynamic one. By solving the SSSP on the fixed graph they try to approximate this path in the dynamic one. We adopt this idea but also take *local* information about graph operations into account allowing us to dynamically assign lower weights to edges in the fixed graph for solving the SSSP. This allows our algorithm to adapt the path on the static graph according to the locally known changes in the dynamic one.

The key idea behind [9] is to estimate the topology of the dynamic graph by using information about the neighboring vertices. More precisely they try to estimate the position of neighboring vehicles by using their respective velocity and direction vector. Our approach also assumes certain knowledge, e.g. the route on the map a vehicle is driving along, about the surrounding topology of a given vertex (see Section 3) thus allowing us more accurate position information since we can also take the curvature of a street into account by computing the position along it.

3 Shortest Path Computation under Unknown Global Graph Operations

Let $G = (V, E)$ be a weighted, directed graph, which is fixed, i.e., no insertions and deletions are allowed. The topology of G is known to the algorithm. Furthermore, $\tilde{G} = (\tilde{V}, \tilde{E})$ denotes a fully dynamic graph. $\tilde{N}$ is a list containing all vertices, which are reachable from a vertex $\tilde{v} \in \tilde{G}$ with distance 1 (direct connection by one edge). Only the vertices in $\tilde{N}$ are known to the algorithm. The remaining topology of $\tilde{G}$ is hidden: Neither the number of vertices is known nor the edges between them. In order to give the reader a clear understanding of these notions, Figure 1 exemplifies these terms when applied to the domain of vehicular ad-hoc networks: Our algorithm is running on green Vehicle 1. Black points mark vertices $v \in G$, here: junctions. Connecting lines in between mark edges $e \in G$ (streets). All vehicles in light blue are located within transmission range and thus are elements of list $\tilde{N}$. Vehicles form vertices of dynamic graph $\tilde{G}$. Dotted lines between them mark edges $\tilde{e} \in \tilde{G}$, e.g. wireless connections. Note, that there is no (direct) edge between vehicle 1 and black vehicles 6,7 and 8 since they are out of transmission range.

We define a set of functions that can be queried by every vertex in $\tilde{G}$:

1. A function $\tilde{f} : \tilde{V} \times T \rightarrow E \times \mathbb{R}^+$ maps a vertex $\tilde{v} \in \tilde{G}$ to exactly one edge $e \in E$. Moreover, it generates a virtual edge between the vertex $\tilde{v}$ and the start vertex of

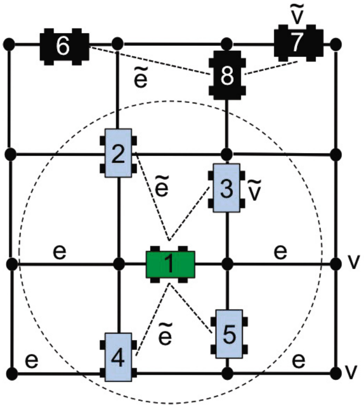


Fig. 1. Notations used throughout the paper on the example of vehicular ad-hoc networks

the assigned edge e and sets a weight depending on the specific application domain. For example, we can interpret this as the distance vehicle $\tilde{v}$ travelled along a road e. $\tilde{f}$ updates this mapping and weight in specific time intervals thus enabling the algorithm to see a snapshot of the neighboring topology of $\tilde{G}$ at a point in time $t \in T$. Hence, a request to $\tilde{f}$ requires a time $t \in T$. Note, that a vertex $\tilde{v} \in \tilde{G}$ is only allowed to query vertices in $\tilde{N}$, which contains direct neighbors within transmission range. Furthermore, queries to future edge weights return null. Also note, that the dynamic graph changes very quickly and $\tilde{f}$ only provides a snapshot of the neighboring topology of $\tilde{G}$.

2. A function $w : V \times \tilde{V} \times T \rightarrow \mathbb{R}^+$ returns a weight of a virtually drawn edge between a vertex in $\tilde{G}$ and a vertex in G. Only queries to elements in $\tilde{N}$ are allowed since the knowledge in a vertex $\tilde{v} \in \tilde{V}$ is restricted to its direct neighbors. In contrast to the virtual edge in $\tilde{f}$ for vehicular ad-hoc networks this gives us the euclidian distance of a vehicle $\tilde{v} \in \tilde{V}$ to a crossing $v \in V$.
3. A function $\tilde{g} : \tilde{V} \rightarrow List < E >$ maps every $\tilde{v} \in \tilde{V}$ to a sequence of edges $e_1, ..., e_n \in E$ effectively describing the path of $\tilde{v}$ on G. In vehicular ad-hoc networks this maps to the most probable path of a vehicle.

After solving the SSSP on fixed graph G yielding shortest path P, the algorithm's task is now to approximate P in $\tilde{G}$ by making use of the information available from the defined functions $\tilde{f}$,w and $\tilde{g}$. In the following, we denote the approximated shortest path as $\tilde{P}$. Note, that P is an array of vertices in G (in ascending order).

Our algorithm is divided into seven steps, which are summarized in Table 1. First, we check if the destination vertex $\tilde{d} \in \tilde{V}$ is a direct neighbor, e.g. an element in $\tilde{N}$. In this case, no shortest path computation is necessary and the algorithm terminates after adding $\tilde{v}$ to $\tilde{P}$.

If the destination vertex is not contained in $\tilde{N}$, we proceed with solving SSSP on G using a modified Dijkstra: Our Dijkstra implementation takes the number of mappings of all $\tilde{v} \in \tilde{N}$ by $\tilde{g}$ to edges in G into account (as far this can be determined by the vertices contained in $\tilde{N}$ and queries to $\tilde{g}$). It assigns lower weights on edges in G where

Table 1. High level steps and actions of the search algorithm

#	Step	Action(s)
1	Check necessity of Single Source Shortest Path computation	Check if a $\tilde{v} \in \tilde{N}$ is a destination. Add $\tilde{v}$ to $\tilde{P}$ and terminate if true or continue otherwise
2	Solve single-source shortest path problem on G	Change weights of edges in G according to number of mappings of vertices $\in \tilde{N}$ to edges $\in G$. Use Dijkstra's Algorithm to compute the shortest path from source to destination
3	Update neighbor weights	Calculate estimated weights $\forall \tilde{v} \in \tilde{N}$ to the start vertex of their mapped edges in G
4	Calculate edge weights between $v \in P$ and $\tilde{v} \in \tilde{N}$	Prefer vertices with minimal edge weight between virtually drawn edges between vertices $v \in P$ and vertices $\tilde{v} \in \tilde{N}$. Weight the result with predefined α
5	Calculate mappings of $\tilde{v} \in \tilde{N}$ with with edges in P	Prefer vertices whose mapping by $\tilde{f}$ is close to edges in P over others. Weight the result with predefined β
6	Calculate edge weights along P	Prefer vertices in with larger sum of edge weights on P before others. Weight the result with predefined γ
7	Update $\tilde{P}$	Select a vertex $\tilde{v} \in \tilde{N}$ according to computed heuristics in steps 4 – 6 and add it to $\tilde{P}$ or wait until next update from f to $\tilde{N}$ if no vertex can be selected. Repeat steps 1 – 7 until destination reached

density of mappings is higher. This is done before every vertex decision for $\tilde{P}$ to accommodate fast topology changes within the graph.

We then estimate future edge weights of all vertices in $\tilde{N}$ to the start vertices of their mapped edge in G. This is especially necessary at low update rates by $\tilde{f}$ to the vertices in $\tilde{N}$. Since our application domain are vehicular ad-hoc networks we have to consider this fact. A car traveling by 36 m/s on a motorway covers a substantial amount of a car's transmission range between two position updates. Hence an update by $\tilde{f}$ could delete an edge in $\tilde{G}$.

The following three steps are crucial for the algorithm since the decision on next vertex for $\tilde{P}$ depends on them. Every vertex in $\tilde{N}$ gets assigned three heuristics within the interval of $[0, 1]$ where higher values denote higher priority in selecting the next vertex for $\tilde{P}$. The three heuristics are given in Table 1, rows 4 – 6.

Each heuristic is multiplied by a weight of α, β and γ respectively. The computations of the three heuristics are described in detail below. The final step is to select a vertex

Algorithm 1. Estimate future edge weights

```
1:  for all ($\tilde{v} \in \tilde{N}$) do
2:      $w_t \leftarrow \tilde{f}(\tilde{v}, t).w$
3:      $w_{t-1} \leftarrow \tilde{f}(\tilde{v}, t_{t-1}).w$
4:      if ($\tilde{f}(\tilde{v}, t).e == \tilde{f}(\tilde{v}, t_{t-1}).e$) then
5:          $\Delta w \leftarrow w_t - w_{t-1}$
6:          $w_{est} \leftarrow w_t + \Delta w$
7:      else
8:          $w_{t-1} \leftarrow \tilde{f}(\tilde{v}, t_{t-1}).e.w - \tilde{f}(\tilde{v}, t_{t-1}).w$
9:          $\Delta w \leftarrow w_t + w_{t-1}$
10:         $w_{est} \leftarrow w_t + \Delta w$
11:     end if
12:     if ($w_{est} > \tilde{f}(\tilde{v}, t).e.w$) then
13:         $E \leftarrow \tilde{g}(\tilde{v})$
14:         for ($i \leftarrow 0; i < E.length - 1; i++$) do
15:             if ($E[i] == \tilde{f}(\tilde{v}, t).e$) then
16:                 $\tilde{v}.e \leftarrow E[i+1]$
17:                 $\tilde{v}.w \leftarrow w_{est} - E[i].w$
18:             end if
19:         end for
20:     else
21:         $\tilde{v}.w \leftarrow w_{est}$
22:     end if
23: end for
```

in $\tilde{N}$ for $\tilde{P}$ based on the heuristics computed in the steps before or restart the algorithm on the vertex until we have have reached a vertex close to destination vertex d.

Estimating Future Weights. The procedure for estimating future weights is given in Algorithm 1. Since $\tilde{f}$ updates $\tilde{N}$ only in specific intervals and the topology of $\tilde{G}$ changes very quickly we try to estimate the topology of the neighboring vertices in $\tilde{G}$. The idea is to query $\tilde{f}$ on the edge weights of all vertices contained in $\tilde{N}$ at time t and $t-1$. We then use the delta of the two weights and add it to the current weight of a vertex in $\tilde{N}$. We have to distinguish two cases here: (1) At time t and time t_{t-1} vertex $\tilde{v}$ maps to the same edge in G, (2) At time t and time t_{t-1} vertex $\tilde{v}$ maps to different edge in G (line 4). This influences how Δw is calculated (lines 5 – 10).

If the estimated edge weight is larger than the weight of the currently assigned edge of $\tilde{v}$ we do not only estimate the weight, we also set the future assigned edge of $\tilde{v}$ using $\tilde{g}$.
We store the updated values directly in $\tilde{v}$.

Heuristic 1: Edge Weights between $v \in P$ and $\tilde{v} \in \tilde{N}$. Algorithm 2 shows how the first heuristic is computed for selecting the next vertex for $\tilde{P}$. The idea here is to use w in order to identify the vertex $\tilde{v} \in \tilde{N}$ with minimum weight to a vertex $v \in P$. It is obvious to prefer these edges for $\tilde{P}$ since they come close to the initial found path P.

The code is executed for $\forall \tilde{v} \in \tilde{N}$ from the previous step and computes $\forall v \in P$ the weight of an edge between a vertex $\tilde{v} \in \tilde{G}$ to a vertex $v \in G$ (line 4-7). We only consider vertices up to a certain weight w_{max}. (lines 8-10). Heuristic 1 is then defined as $m_1 = \alpha * (1 - \frac{w}{w_{max}})$ (line 11). Remember, that higher values denote higher priority in selecting the next vertex for $\tilde{P}$. Heuristic 1 is optimizing the stability of the path $\tilde{P}$ since it prefers vertices $\tilde{v} \in \tilde{V}$ that are close to the path vertices in P. In vehicular ad-hoc networks we can interpret this as preferring vehicles that are on street junctions which form natural turning points in the street graph.

Algorithm 2. Heuristic 1: Edge weights between $v \in P$ and $\tilde{v} \in \tilde{N}$

```
 1: for all $(\tilde{v} \in \tilde{N})$ do
 2:     $w \leftarrow \infty$
 3:     for all $(v \in P)$ do
 4:         if $(w(v, \tilde{v}, t) < w)$ then
 5:             $w \leftarrow w(v, \tilde{v}, t)$
 6:         end if
 7:         if $(w > w_{max})$ then
 8:             $w \leftarrow w_{max}$
 9:         end if
10:     end for
11:     $\tilde{v}.h_1 \leftarrow \alpha * (1 - \frac{w}{w_{max}})$
12: end for
```

Algorithm 3. Heuristic 2: Mappings of $\tilde{v} \in \tilde{N}$ with with edges in P

```
for all $(\tilde{v} \in \tilde{N})$ do
    $currentScore \leftarrow 0$
    for $(i \leftarrow 0; i < P.size - 2; i++)$ do
        $p_1 \leftarrow P[i]$
        $p_2 \leftarrow P[i+1]$
        if $(edge(p_1, p_2) \in \tilde{g}(\tilde{v}))$ then
            $currentScore \leftarrow currentScore + \omega$
        else
            if $(edge(p_2, p_1) \in \tilde{g}(\tilde{v}))$ then
                $currentScore \leftarrow currentScore + \tau$
            end if
        end if
    end for
    $\tilde{v}.h_2 \leftarrow \beta * (\frac{currentScore}{score_{max}})$
end for
```

Heuristic 2: Mappings of $\tilde{v} \in \tilde{N}$ with Edges in P. Heuristic 2 is computed as stated in algorithm 3. As for heuristic 1, the code is executed $\forall \tilde{v} \in \tilde{N}$. It scores vertices in $\tilde{N}$ higher whose mappings to edges in G matches more edges in P over others. Furthermore, we take the direction of the edge into account: An exact match gets assigned a score of ω. If the mapped edge of $\tilde{v}$ is the opposite of an edge in P we assign τ.

If no match is detected, we assign a score of 0. Note, that we require $\omega > \tau$. The actual values used for evaluation are given in table 2.

$Score_{max}$ is defined as $\sum_{i=0}^{|P|-1} \omega$. Heuristic 2 is defined as $\beta * \frac{score}{score_{max}}$, yielding to a value in the range $[0, 1]$ where larger values denote higher priority in selecting the next vertex for $\tilde{P}$. Finally, we assign the weight β and store heuristic h_2 in $\tilde{v}$ (line 14). Heuristic 2 is trying to optimize path stability by preferring $\tilde{v} \in \tilde{V}$ whose own path along G covers more of path P, the idea being that if $\tilde{v}$ cannot find next vertex for $\tilde{P}$ it at least gets closer to the destination. In vehicular ad-hoc network terms we can interpret this as preferring a vehicle that can carry the message closer to the destination in the case when a more suitable vehicle cannot be found.

Algorithm 4. Heuristic 3: Edge weights along P

```
$w_P \leftarrow P.totalWeight$
for all $(\tilde{v} \in \tilde{N})$ do
    $e_{\tilde{v}} \leftarrow \tilde{f}(\tilde{v}, t).e$
    $w \leftarrow 0$
    if $(e_{\tilde{v}} \in P)$ then
        for $(i \leftarrow 0; i < P.size - 2; i++)$ do
            if $(e_{\tilde{v}} == edge(P[i], P[i+1]))$ then
                $w \leftarrow w + \tilde{f}(\tilde{v}, t).w$
                $break$
            else
                $w \leftarrow w + edge(P[i], P[i+1]).w$
            end if
        end for
    else
        $leastWeight \leftarrow \infty$
        $v \leftarrow nil$
        for $(i \leftarrow 0; i < P.size - 1; i++)$ do
            if $(w(P[i], \tilde{v}, t) < leastWeight)$ then
                $leastWeight \leftarrow w(P[i], \tilde{v}, t)$
                $v \leftarrow P[i]$
            end if
        end for
        for $(i \leftarrow 0; i < P.size - 2; i++)$ do
            if $(P[i] == v)$ then
                $break$
            else
                $w \leftarrow w + edge(P[i], P[i+1]).w$
            end if
        end for
    end if
    $\tilde{v}.h_3 \leftarrow \gamma * (\frac{w}{w_P})$
end for
```

Heuristic 3: Edge Weights along P. The pseudocode for the third heuristic is given in Algorithm 4. It favors vertices in $\tilde{N}$ with larger sum of edge weights on P before others. We first look at the current mapping of vertex $\tilde{v} \in \tilde{N}$ to an edge in G and distinguish two cases:

- P **Contains the Current Mapping.** We add up the weight of edges $e \in P$ starting from $P[0]$ to the current mapping of $\tilde{v}$ (lines 5-13)
- P **doesn't Contain the Current Mapping.** We add up the weight of every edge $e \in P$ until the vertex with the least weight from $\tilde{v}$ to a vertex $v \in P$ (lines 15-30)

Like before, we calculate a heuristic in the interval (0,1) We multiply by a weight of γ (line 31). Also here, larger values denote higher priority in selecting the next vertex for $\tilde{P}$. In contrast to heuristic 1 and 2 this heuristics tries to optimize progress towards the destination by choosing the next vertex $\tilde{v} \in \tilde{V}$ that is furthest along P. In vehicular ad-hoc networks this means choosing the vehicle furthest along the guiding path on the street map.

Selecting the Next Vertex for $\tilde{P}$. After calculating the heuristics described in the previous sections we can now select a vertex in $\tilde{N}$ for $\tilde{P}$. Let H be the set containing the sums of h_1, h_2, h_3 $\forall n \in \tilde{N}$. Then, the next vertex for $\tilde{P}$ is defined as the vertex with the largest sum in H.

If the next vertex is $\tilde{P}[\tilde{P}.length - 1]$ we re-calculate after the next update of $\tilde{f}$.

4 Evaluation

We integrated our algorithm into the transportation layer of a network stack and performed a simple point-to-point sending task in a simulator for evaluation of vehicular ad-hoc network applications (V2X Simulation Runtime Infrastructure or short VSimRTI [10]). VSimRTI integrates and coordinates different simulators and constitutes a middleware between the individual simulators. For realistic simulation we used the wireless network simulator Jist/Swans [11] and the traffic simulation SUMO [12]. We furthermore integrated the effect of buildings on wireless transmission into the simulator. Two vehicles are only able to communicate if and only if there is a line of sight between them. This serves as a lower bound on the connection between the vehicles in the network simulator.

We compare the performance of our algorithm with SEET [8] and GRANELLI [9] in terms of path discovery ratio (PDR) and path discovery time (PDT). The first metric measures the ratio
$PDR = \frac{SuccessfulShortestPathDiscoveries}{TotalShortestPathSearches}$
and the second indicates time from beginning to end of shortest path search $PDT = t_{end} - t_{start}$.

4.1 Implementation of $\tilde{f}$, w and $\tilde{g}$

Remember that graph $G = (V, E)$ is represented by the underlying city map. Junctions are vertices $v \in V$, road segments denote edges $e \in E$. $\tilde{G} = (\tilde{V}, \tilde{E})$ is spanned by the

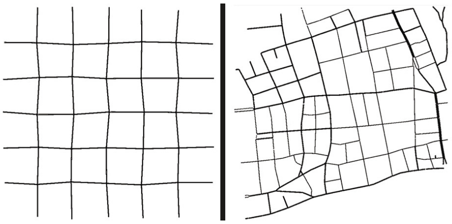


Fig. 2. Fixed graphs used for parameter optimization and evaluation. Left: Random generated, used for parameter optimization (1200m x 1200m). Right: Graph generated based on an existing road network of the city of Heidelberg, Germany used for evaluation (1200m x 1200m).

Table 2. Values of the various weights used during the evaluation. The three on the left side denote the weights for the different heuristics while the two on the right were used for score computation of Heuristic 2 (see Algorithm 3).

Heuristic Weight		Heuristic 2 score	
weight	value	score	value
α	0.5	ω	2
β	0.3	τ	1
γ	0.2		

vehicular ad-hoc network. Vehicles are represented by $\tilde{v} \in \tilde{V}$, edges $\tilde{e} \in \tilde{E}$ are considered as wireless connection between two vehicles. The abstract defined functions $\tilde{f}$, w and $\tilde{g}$ of Section 3 are then implemented as follows: Given a time t, $\tilde{f}$ assigns vehicles to specific road segments. Weight is calculated out of the distance to the beginning of the assigned road segment. In our application domain, $\tilde{f}$ is responsible for controlling and updating positions of vehicles therefore realizing vehicle movements over time. Function w returns the distance between a vehicle in transmission range and a junction of the city map. The vehicles further include the road segments which they have passed as well as their most probable path in their position updates. This realizes function $\tilde{g}$.

4.2 Scenario

Evaluation was done in a scenario where every vehicle starts one shortest path computation to a given car, e.g. the center of the underlying city map. The destination car remained stationary while all others were driving a route on the map.

We optimized weights for heuristics 1 – 3 introduced in Section 3 on a randomly generated map shown in Figure 2 (left). Weights and values for ω and τ, which were found to be optimal for our algorithm, are given in Table 2. We optimized for high PDR.

After optimization, evaluation was done on a map generated out of an existing city environment (Heidelberg, Germany, Figure 2, right). Evaluation runtime was 120 seconds where we started shortest path computation after 20 seconds simulation time. This ensured a fair distribution of vehicles on the map. n vehicles per second were placed on the map by the simulator and removed after they completed their route where

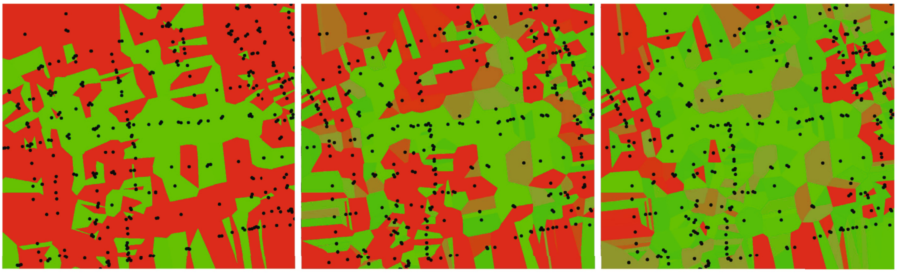


Fig. 3. Voronoi diagrams visualizing PDR and PDT. Dots mark nodes in which path computation started, color denotes average time until shortest path completed in the enclosed region. Left: Granelli ($n = 4$), fast but unreliable; middle: Seet ($n = 4$), slower but more reliable than Granelli due to taking correlation of $\tilde{G}$ and G into account; right: Our approach, $n = 4$, taking correlation between $\tilde{G}$ and G as well as local information on dynamic graph operations into account outperforms Granelli and Seet.

$n \in \{1, 2, 3, 4\}$. For $n = 1$ this resulted in 120 path computations. We repeated every run three times for every n and algorithm. This means $120 * 3 = 360$ path computations for $n = 1$ in total per algorithm and 720, 1080, 1440 for $n = 2, 3, 4$ respectively resulting in a total of 3600 path searches per algorithm.

4.3 Results and Analysis

Results in Figure 4 and Figure 5 show that our approach outperforms Seet and Granelli in means of PDR. As expected, PDR increases with increasing traffic density. Seet obviously benefits from the available information about the underlying city map (the static graph G). Granelli lacks this kind of information which results in a

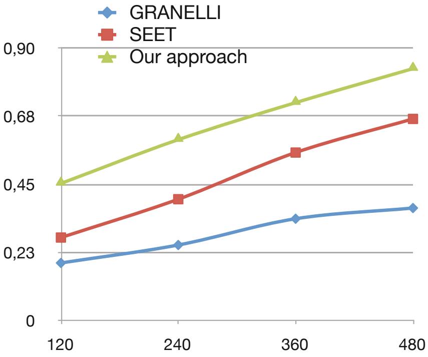


Fig. 4. Path Discovery Ratio (PDR) results for all three algorithms. X-Axis gives the number of path searches, y-axis gives the percentage of successful path searches. As expected, PDR increases with a larger number of cars on the evaluation scenario.

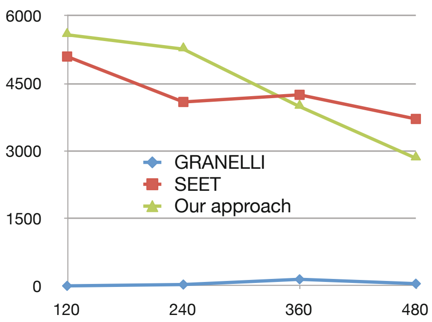
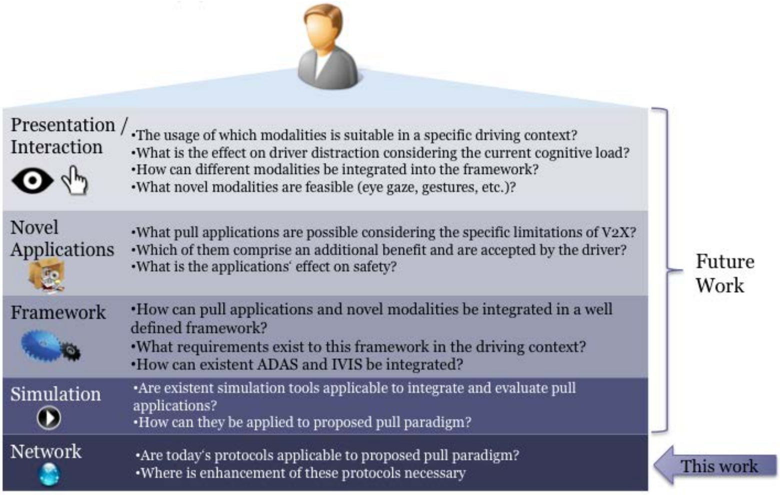


Fig. 5. Path Discovery Time (PDT) for all three algorithms. X-Axis gives the number of path searches, y-axis gives the time in ms. GRANELLI does not reschedule when no suitable successor vertex for the shortest path can be found. By increasing the number of vertices in the graph, PDT gets lower for SEET and our approach.



Presentation / Interaction
- The usage of which modalities is suitable in a specific driving context?
- What is the effect on driver distraction considering the current cognitive load?
- How can different modalities be integrated into the framework?
- What novel modalities are feasible (eye gaze, gestures, etc.)?

Novel Applications
- What pull applications are possible considering the specific limitations of V2X?
- Which of them comprise an additional benefit and are accepted by the driver?
- What is the applications' effect on safety?

Framework
- How can pull applications and novel modalities be integrated in a well defined framework?
- What requirements exist to this framework in the driving context?
- How can existent ADAS and IVIS be integrated?

Simulation
- Are existent simulation tools applicable to integrate and evaluate pull applications?
- How can they be applied to proposed pull paradigm?

Network
- Are today's protocols applicable to proposed pull paradigm?
- Where is enhancement of these protocols necessary

Fig. 6. Context of this work: Embedded in a framework for developing novel V2X enabled applications

lower PDR. As our approach also takes local information about graph operations in $\tilde{G}$ (the vehicular ad-hoc network) into account, it scores higher PDR than both SEET and GRANELLI. The results are statistically significant ($p < .001$ according to a $\chi^2 test$).

However, by means of PDT, GRANELLI is superior to SEET and our approach. Both, SEET and our approach re-schedule path searching in a vertex when no suitable successor vertex could be identified for the shortest path (see Section 3). GRANELLI's behavior in such a situation is to greedily select a next vertex out of the neighboring vertices and do no re-scheduling at all. This also justifies low PDR for GRANELLI. Interestingly, PDT increases for GRANELLI but decreases for both SEET and our approach when there are more vehicles on the graph. In this case both, SEET and our approach have an increased number of vertices available for choosing the next vertex for the shortest path and the probability of finding a suitable one also increases since re-computing time for heuristics is lower than the re-scheduling interval PDT decreases in this case. Because our approach considers local information about graph operations it supersedes SEET by means of PDT over time which results in a lower number of re-schedules.

Figure 3 depicts voronoi visualizations for all three algorithms after a run with $n = 4$. Black dots mark starting positions for shortest path computation. Enclosing colored areas denote PDT to the centre of the map for an individual path in ms (more red areas mark higher PDT). After a threshold of 15000 ms path computation was stopped and marked as *failed*. One clearly recognizes short PDT of GRANELLI but low PDR: Paths are found quickly or not at all. Re-scheduling in cases when no successor vertex for the shortest path can be identified results in stepwise PDTs. Results of our approach reflect high PDR even for large path lengths due to exploiting local information on dynamic graph operations.

5 Conclusions

In this paper, we developed a combined uniform and heuristic search algorithm for maintaining shortest paths in fully dynamic graphs. While other approaches assume global knowledge on performed graph operations, we argued that there exist use cases where this information is not available. Our approach shows that in those cases the algorithms' performance can greatly benefit from considering domain specific knowledge. In our example, we instantiated two graphs: A static and dynamic one. We exploited domain specific relations between these graphs in order to heuristically maintain a shortest path in a dynamic graph. The used heuristics are also tailored to the domain.

We applied our approach to vehicular ad-hoc networks and integrated it into the transportation layer of a network stack to use it for routing data packets between two vehicles. Evaluation was performed against two other routing algorithm of this domain. Due to re-scheduling when no neighboring vertex could be identified during shortest path search, the approach of GRANELLI is superior to our implementation in means of PDT. However, our approach outperformed SEET and GRANELLI in means of PDR.

6 Future Work

The algorithm presented in this work is able to maintain shortest paths on highly dynamic graphs. The goal of this paper was to solve DSSSP without global knowledge of graph topology. We therefore made the assumption that the goal vertex is given to the algorithm. In the future we plan to omit this assumption and enhance the algorithm presented here in a way that the actual search for the goal vertex is part of the task. Moreover, we want to discover the "least weighted circuit" between start and goal vertex. This additional constraints are significantly more challenging than DSSSP since topology is constantly changing. Most likely, the path back to the start vertex is not equal to the path on the forward run. We are currently tackling this problem by exploiting swarm intelligence: Successful discovered “Trails” are stored and updated in each vertex visited during shortest path computation. Another search will then be able to reuse information from previous successful attempts.

While we believe our algorithm is generic, we will integrate our algorithm into a larger system for developing and evaluating novel V2X applications (see Figure 6). This work is part of the network layer in Figure 6. Current network protocols are intended to broadcast (mostly) safety related information as quickly as possible over the ad-hoc network. When the driver or the system requests information available somewhere in the network, the task gets more challenging. This addresses questions on how to find the relevant information in a highly dynamic ad-hoc network and how to route back information once it is found. It also addresses dissemination and caching of information within the ad-hoc network. We approach these challenges by transferring them to graph theory and use results from this field of research as well as from the target domain in order to combine successful approaches from both resulting in a more performant system.

Large scale field tests for testing and evaluating V2X technology and its applications are expensive and potentially dangerous. This is why a lot of research in the past has focused on the simulation of the underlying network communication in order to test specific use-cases in the lab before they get integrated into real cars. To simulate a use-case, like a local danger warning for example, it requires more than one simulator: Network simulation, traffic simulation and environment simulation have to be coordinated. This work is done in simulation layer as depicted in Figure 6.

V2X technology itself constitutes enormous potential for another kind of use-case beyond todays approach. Up to now, information in V2X projects is primarily “pushed” from sender to receiver in order to disseminate messages and warnings over the ad-hoc network. The driver remains more or less passive in this process. Value-added services like parking or urban information are not based on V2V in today’s paradigm. These use-cases rely on mobile connections (UMTS). Other interactive applications like enabling voice chat between cars on the basis of V2X are actually not a novel approach, but only a different technical realization of an established use-case (communication between truck drivers). Providing V2X technology to the drivers could, however, largely increase the acceptance and open a wide range of additional use-cases. Here, many questions also remain unaddressed so far. What applications are possible with this technology being available to the driver? Which of them are accepted by and useful for drivers? What is the applications effect on safety? The usage of which modalities is suitable in a specific driving context? The answer could be different depending on the current traffic situation (stationary vs. moving vehicle).

The framework layer in Figure 6 serves as a basis for addressing the human factor issues discussed above that are related to V2X applications. At the same time, it considers the specific limitations and requirements of the V2X technology. Using this platform, traditional and novel applications for V2X technology will be developed and evaluated. Novel in the sense, that we open the V2X technology to the user. This will go beyond established “Push-Applications” and introduce a paradigm shift to “Pull-Applications”.

References

1. Dijkstra, E.W.: A note on two problems in connection with graphs. Numerische Mathematik 1, 269–271 (1959)
2. Nannicini, G., Liberti, L.: Shortest paths on dynamic graphs (2008)
3. Koenig, S., Likhachev, M., Furcy, D.: Lifelong planning A*. Artif. Intell. 155, 93–146 (2004)
4. Misra, S., Oommen, B.J.: Stochastic Learning Automata-Based Dynamic Algorithms for the Single Source Shortest Path Problem. In: Orchard, B., Yang, C., Ali, M. (eds.) IEA/AIE 2004. LNCS (LNAI), vol. 3029, pp. 239–248. Springer, Heidelberg (2004)
5. Cicerone, S., Stefano, G.D., Frigioni, D., Nanni, U.: A fully dynamic algorithm for distributed shortest paths. Theoretical Computer Science 297, 1–3 (2003)
6. Demetrescu, C., Italiano, G.F.: A new approach to dynamic all pairs shortest paths. In: Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing, STOC 2003, pp. 159–166. ACM, New York (2003)
7. Frigioni, D., Spaccamela, A.M., Nanni, U.: Fully dynamic algorithms for maintaining shortest path trees. Algorithms 34, 251–281 (2000)
8. Seet, B.-C., Liu, G., Lee, B.-S., Foh, C.-H., Wong, K.-J., Lee, K.-K.: A-STAR: A Mobile Ad Hoc Routing Strategy for Metropolis Vehicular Communications. In: Mitrou, N.M., Kontovasilis, K., Rouskas, G.N., Iliadis, I., Merakos, L. (eds.) NETWORKING 2004. LNCS, vol. 3042, pp. 989–999. Springer, Heidelberg (2004)
9. Granelli, F., Boato, G., Kliazovich, D., Vernazza, G.: Enhanced gpsr routing in multi-hop vehicular communications through movement awareness. IEEE Communications Letters 11, 781–783 (2007)
10. Naumann, N., Schiinemann, B., Radusch, I.: Vsimrti - simulation runtime infrastructure for v2x communication scenarios. In: Proceedings of the 16th World Congress and Exhibition on Intelligent Transport Systems and Services (2009)
11. Barr, R., Haas, Z.J., van Renesse, R.: Jist: An efficient approach to simulation using virtual machines. Software Practice & Experience 35, 539–576 (2005)
12. Krajzewicz, D., Hertkorn, G., Rössel, C., Wagner, P.: Sumo (simulation of urban mobility); an open-source traffic simulation. In: Proceedings of the 4th Middle East Symposium on Simulation and Modelling (2002)

Admissible Distance Heuristics for General Games

Daniel Michulke[1] and Stephan Schiffel[2]

[1] Dresden University of Technology, Dresden, Germany
[2] Reykjavík University, Reykjavík, Iceland
dmichulke@gmail.com, stephans@ru.is

Abstract. A general game player is a program that is able to play arbitrary games well given only their rules. One of the main problems of general game playing is the automatic construction of a good evaluation function for these games. Distance features are an important aspect of such an evaluation function, measuring, e.g., the distance of a pawn towards the promotion rank in chess or the distance between Pac-Man and the ghosts.

However, current distance features for General Game Playing are often based on too specific detection patterns to be generally applicable, and they often apply a uniform Manhattan distance regardless of the move patterns of the objects involved. In addition, the existing distance features do not provide proven bounds on the actual distances.

In this paper, we present a method to automatically construct distance heuristics directly from the rules of an arbitrary game. The presented method is not limited to specific game structures, such as Cartesian boards, but applicable to all structures in a game. Constructing the distance heuristics from the game rules ensures that the construction does not depend on the size of the state space, but only on the size of the game description which is exponentially smaller in general. Furthermore, we prove that the constructed distance heuristics are admissible, i.e., provide proven lower bounds on the actual distances.

We demonstrate the effectiveness of our approach by integrating the distance heuristics in an evaluation function of a general game player and comparing the performance with a state-of-the-art player.

Keywords: General game playing, Feature construction, Heuristic search.

1 Introduction

While in classical game playing, human experts encode their knowledge into features and parameters of evaluation functions (e.g., weights), the goal of General Game Playing is to develop programs that are able to autonomously derive a good evaluation function for a game given only the rules of the game. Because the games are unknown beforehand, the main problem lies in the detection and construction of useful features and heuristics for guiding search in the match.

One class of such features are distance features used in a variety of GGP agents (e.g., [6,9,2,4]). The way of detecting and constructing features in current game playing systems, however, suffers from a number of disadvantages:

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 188–203, 2013.

- Distance features require a prior recognition of board-like game elements. Current approaches formulate hypotheses about which element of the game rules describes a board and then either check these hypotheses in internal simulations of the game (e.g., [6,9,4]) or try to prove them [10]. Both approaches are expensive and can only detect boards if their description follows a certain syntactic pattern.
- Distance features are limited to Cartesian board-like structures, that is, n-dimensional structures with totally ordered coordinates. Distances over general graphs are not considered.
- Distances are calculated using a predefined metric on the boards. Consequently, distance values obtained do not depend on the type of piece involved. For example, using a predefined metric the distance of a rook, king and pawn from $a2$ to $c2$ would appear equal while a human would identify the distance as 1, 2 and ∞ (unreachable), respectively.

In this paper we will present a more general approach for the construction of distance features for general games. The underlying idea is to analyze the rules of game in order to find dependencies between the fluents of the game, i.e., between the atomic properties of the game states. Based on these dependencies, we define a distance function that computes an admissible estimate for the number of steps required to make a certain fluent true. This distance function can be used as a feature in search heuristics of GGP agents. In contrast to previous approaches, our approach does not depend on syntactic patterns and involves no internal simulation or detection of any predefined game elements. Moreover, it is not limited to board-like structures but can be used for every fluent of a game.

The remainder of this paper is structured as follows: In the next section we give an introduction to the Game Description Language (GDL), which is used to describe general games. In Section 3 we introduce the theoretical basis for this work, so called fluent graphs, and show how to use them to derive distances from states to fluents. We proceed in Section 4 by showing how fluent graphs can be constructed from a game description and demonstrate their application in Section 5. We conduct experiments in Section 6 to show the benefit and generality of our approach and discuss related approaches in Section 7. Finally, we give an outlook on future work in Section 8 and summarize in Section 9.

2 Preliminaries

The language used for describing the rules of general games is the Game Description Language [7] (GDL). GDL is an extension of Datalog with functions, equality, some syntactical restrictions to preserve finiteness, and some predefined keywords.

The following is a partial encoding of a Tic-Tac-Toe game in GDL. In this paper we use Prolog syntax where words starting with upper-case letters stand for variables and the remaining words are constants.

```
1  role(xplayer). role(oplayer).
2
3  init(control(xplayer)).
4  init(cell(1,1,b)). init(cell(1,2,b)).init(cell(1,3,b)).
5   ...
6  init(cell(3,3,b)).
7
8  legal(P, mark(X,Y)) :- true(control(P)), true(cell(X,Y,b)).
9  legal(P, noop) :- role(P), not true(control(P)).
10
11 next(cell(X,Y,x)) :- does(xplayer, mark(X,Y)).
12 next(cell(X,Y,o)) :- does(oplayer, mark(X,Y)).
13 next(cell(X,Y,C)) :- true(cell(X,Y,C)), distinct(C, b).
14 next(cell(X,Y,b)) :- true(cell(X,Y,b)), does(P, mark(M,N)),
15   (distinct(X,M) ; distinct(Y,N)).
16
17 goal(xplayer, 100) :- line(x).
18 ...
19 terminal :- line(x) ; line(o) ; not open.
20
21 line(P) :-
22   true(cell(X,1,P)), true(cell(X,2,P)), true(cell(X,3,P)).
23 ...
24 open :- true(cell(X,Y,b)).
```

The first line declares the roles of the game. The unary predicate **init** defines the properties that are true in the initial state. Lines 8-9 define the legal moves of the game with the help of the keyword **legal**. For example, mark(X,Y) is a legal move for role P if control(P) is true in the current state (i.e., it's P's turn) and the cell X,Y is blank (cell(X,Y,b)). The rules for predicate **next** define the properties that hold in the successor state, e.g., cell(M,N,x) holds if xplayer marked the cell M,N and cell(M,N,b) does not change if some cell different from M,N was marked[1]. Lines 17 to 19 define the rewards of the players and the condition for terminal states. The rules for both contain auxiliary predicates line(P) and open which encode the concept of a line-of-three and the existence of a blank cell, respectively.

We will refer to the arguments of the GDL keywords **init**, **true** and **next** as fluents. Fluents are the building blocks of states in GDL and will be in the center of our analysis for distance features. In the above example, there are two different types of fluents, control(X) with X $\in$ {xplayer, oplayer} and cell(X, Y, Z) with X, Y $\in$ {1, 2, 3} and Z $\in$ {b, x, o}.

In [11], we defined a formal semantics of a game described in GDL as a state transition system:

Definition 1 (Game). *Let Σ be a set of ground terms and 2^Σ denote the set of finite subsets of Σ. A game over this set of ground terms Σ is a state transition system $\Gamma = (R, s_0, T, l, u, g)$ over sets of states $\mathcal{S} \subseteq 2^\Sigma$ and actions $\mathcal{A} \subseteq \Sigma$ with*

[1] The special predicate **distinct**(X,Y) holds if the terms X and Y are syntactically different.

- *$R \subseteq \Sigma$, a finite set of roles;*
- *$s_0 \in \mathcal{S}$, the initial state of the game;*
- *$T \subseteq \mathcal{S}$, the set of terminal states;*
- *$l : R \times \mathcal{A} \times \mathcal{S}$, the legality relation;*
- *$u : (R \mapsto \mathcal{A}) \times \mathcal{S} \to \mathcal{S}$, the transition or update function; and*
- *$g : R \times \mathcal{S} \mapsto \mathbb{N}$, the reward or goal function.*

This formal semantics is based on a set of ground terms Σ. This set is the set of all ground terms over the signature of the game description. Hence, fluents, actions and roles of the game are ground terms in Σ. States are finite sets of fluents, i.e., finite subsets of Σ. The connection between a game description D and the game Γ it describes is established using the standard model of the logic program D. For example, the update function u is defined as

$$u(A, s) = \{f \in \Sigma : D \cup s^{\texttt{true}} \cup A^{\texttt{does}} \models \texttt{next}(f)\}$$

where $s^{\texttt{true}}$ and $A^{\texttt{does}}$ are suitable encodings of the state s and the joint action A of all players as a logic program. Thus, the successor state $u(A, s)$ is the set of all ground terms (fluents) f such that $\texttt{next}(f)$ is entailed by the game description D together with the state s and the joint move A. For a complete definition for all components of the game Γ we refer to [11].

3 Fluent Graphs

Our goal is to obtain knowledge on how fluents evolve over time. We start by building a *fluent graph* that contains all the fluents of a game as nodes. Then we add directed edges (f_i, f) if at least one of the predecessor fluents f_i must hold in the current state for the fluent f to hold in the successor state. Figure 1(a) shows a partial fluent graph for Tic-Tac-Toe that relates the fluents `cell(3,1,Z)` for `Z` $\in$ `{b, x, o}`.

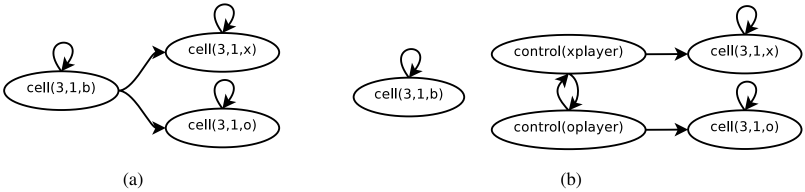


Fig. 1. Shown are two possible partial fluent graphs for Tic-Tac-Toe, where (a) captures the dependencies between different markers on a cell while (b) fails to capture these dependencies

For cell `(3,1)` to be blank it had to be blank before. For a cell to contain an `x` (or an `o`) in the successor state there are two possible preconditions. Either, it contained an `x` (or `o`) before or it was blank.

Using this graph, we can conclude that, e.g., a transition from `cell(3,1,b)` to `cell(3,1,x)` is possible within one step while a transition from `cell(3,1,o)` to `cell(3,1,x)` is impossible.

To build on this information, we formally define a fluent graph as follows:

Definition 2 (Fluent Graph). *Let Γ be a game over ground terms Σ. A graph $G = (V, E)$ is called a fluent graph for Γ iff*

- $V = \Sigma \cup \{\emptyset\}$ *and*
- *for all fluents $f \in \Sigma$, two valid states s and s'*

$$\begin{aligned} &(s' \textit{ is a successor of } s) \wedge f' \in s' \\ &\quad \Rightarrow (\exists f)(f, f') \in E \wedge (f \in s \cup \{\emptyset\}) \end{aligned} \tag{1}$$

In this definition we add an additional node $\emptyset$ to the graph and allow $\emptyset$ to occur as the source of edges. The reason is that there can be fluents in the game that do not have any preconditions, for example the fluent `g` with the following next rule: **`next`**`(g) :-` **`distinct`**`(a,b)`. On the other hand, there might be fluents that cannot occur in any state, because the body of the corresponding next rule is unsatisfiable, for example: **`next`**`(h) :-` **`distinct`**`(a,a)`. We distinguish between fluents that have no precondition (such as g) and fluents that are unreachable (such as h) by connecting the former to the node $\emptyset$ while unreachable fluents have no edge in the fluent graph.

Note that the definition covers only some of the necessary preconditions for fluents, therefore fluent graphs are not unique as Figure 1(b) shows. We will address this problem later.

We can now define a distance function $\Delta(s, f')$ between the current state s and a state in which fluent f' holds as follows:

Definition 3 (Distance Function). *Let $\Delta_G(f, f')$ be the length of the shortest path from node f to node f' in the fluent graph G or ∞ if there is no such path. Then we define*

$$\Delta(f, f') \overset{\text{def}}{=} \begin{cases} 0 & f = f' \\ \Delta_G(f, f') & else \end{cases}$$

$$\Delta(s, f') \overset{\text{def}}{=} \min_{f \in s \cup \{\emptyset\}} \Delta(f, f')$$

That means, the distance $\Delta(s, f')$ is 0 if and only if f' holds in s, otherwise it is compute as the shortest path in the fluent graph from any fluent in s to f'.

Intuitively, each edge (f, f') in the fluent graph corresponds to a state transition of the game from a state in which f holds to a state in which f' holds. Thus, the length of a path from f to f' in the fluent graph corresponds to the number of steps in the game between a state containing f to a state containing f'. Of course, the fluent graph is an abstraction of the actual game: many preconditions for the state transitions are ignored. As a consequence, the distance $\Delta(s, f')$ that we compute in this way is a lower bound on the actual number of steps it takes to go from s to a state in which f' holds. Therefore the distance $\Delta(s, f')$ is an admissible heuristic for reaching f' from a state s.

Theorem 1 (Admissible Distance). *Let*
- $\Gamma = (R, s_0, T, l, u, g)$ *be a game with ground terms Σ and states $\mathcal{S}$,*
- $s_1 \in \mathcal{S}$ *be a state of Γ,*

- *$f \in \Sigma$ be a fluent of Γ, and*
- *$G = (V, E)$ be a fluent graph for Γ.*

Furthermore, let $s_1 \mapsto s_2 \mapsto \ldots \mapsto s_{m+1}$ denote a legal sequence of states of Γ, that is, for all i with $0 < i \leq m$ there is a joint action A_i, such that:

$$s_{i+1} = u(A_i, s_i) \wedge (\forall r \in R) l(r, A_i(r), s)$$

If $\Delta(s_1, f) = n$, then there is no legal sequence of states $s_1 \mapsto \ldots \mapsto s_{m+1}$ with $f \in s_{m+1}$ and $m < n$.

Proof. We prove the theorem by contradiction. Assume that $\Delta(s_1, f) = n$ and there is a legal sequence of states $s_1 \mapsto \ldots \mapsto s_{m+1}$ with $f \in s_{m+1}$ and $m < n$. By Definition 2, for every two consecutive states s_i, s_{i+1} of the sequence $s_1 \mapsto \ldots \mapsto s_{m+1}$ and for every $f_{i+1} \in s_{i+1}$ there is an edge $(f_i, f_{i+1}) \in E$ such that $f_i \in s_i$ or $f_i = \emptyset$. Therefore, there is a path $f_j, \ldots, f_m, f_{m+1}$ in G with $1 \leq j \leq m$ and the following properties:

- $f_i \in s_i$ for all $i = j, ..., m + 1$,
- $f_{m+1} = f$, and
- either $f_j \in s_1$ (e.g., if $j = 1$) or $f_j = \emptyset$.

Thus, the path $f_j, \ldots, f_m, f_{m+1}$ has a length of at most m.

Consequently, $\Delta(s_1, f) \leq m$ because $f_j \in s_1 \cup \{\emptyset\}$ and $f_{m+1} = f$. However, $\Delta(s_1, f) \leq m$ together with $m < n$ contradicts $\Delta(s_1, f) = n$. □

4 Constructing Fluent Graphs from Rules

We propose an algorithm to construct a fluent graph based on the rules of the game. The transitions of a state s to its successor state s' are encoded fluent-wise via the **`next`** rules. Consequently, for each $f' \in s'$ there must be at least one rule with the head **`next`**`(f')`. All fluents occurring in the body of these rules are possible sources for an edge to f' in the fluent graph.

For each ground fluent f' of the game:

1. Construct a ground disjunctive normal form ϕ of `next`(f'), i.e., a formula ϕ such that `next`$(f') \supset \phi$.
2. For every disjunct ψ in ϕ:
   - Pick one literal `true`(f) from ψ or set $f = \emptyset$ if there is none.
   - Add the edge (f, f') to the fluent graph.

Note, that we only select one literal from each disjunct in ϕ. Since, the distance function $\Delta(s, f')$ obtained from the fluent graph is admissible, the goal is to construct a fluent graph that increases the lengths of the shortest paths between the fluents as much as possible. Therefore, the fluent graph should contain as few edges as possible. In general the complete fluent graph (i.e., the graph where every fluent is connected to every other fluent) is the least informative because the maximal distance obtained from this graph is 1.

The algorithm outline still leaves some open issues:

1. How do we construct a ground formula ϕ that is the disjunctive normal form of $\texttt{next}(f')$?
2. Which literal $\texttt{true}(f)$ do we select if there is more than one? Or, in other words, which precondition f' of f do we select?

We will discuss both issues in the following sections.

4.1 Constructing a DNF of $\texttt{next}(\mathbf{f'})$

A formula ϕ in DNF is a set of formulas $\{\psi_1, \ldots, \psi_n\}$ connected by disjunctions such that each formula ψ_i is a set of literals connected by conjunctions. We propose the algorithm in Figure 1 to construct ϕ such that $\texttt{next}(f') \supset \phi$.

Algorithm 1. Constructing a formula ϕ in DNF with $\texttt{next}(f') \supset \phi$

```
Input: game description $D$, ground fluent $f'$
Output: $\phi$, such that $\texttt{next}(f') \supset \phi$
 1: $\phi := \texttt{next}(f')$
 2: $finished := false$
 3: while $\neg finished$ do
 4:    Replace every positive occurrence of $\texttt{does}(r,a)$ in $\phi$ with $\texttt{legal}(r,a)$.
 5:    Select a positive literal $l$ from $\phi$ such that $l \neq \texttt{true}(t), l \neq \texttt{distinct}(t_1,t_2)$ and $l$ is
       not a recursively defined predicate.
 6:    if there is no such literal then
 7:       $finished := true$
 8:    else
 9:       $\hat{l} := \bigvee_{h\,:\text{-}\,b \in D,\, l\sigma = h\sigma} b\sigma$
10:       $\phi := \phi\{l/\hat{l}\}$
11:    end if
12: end while
13: Transform $\phi$ into disjunctive normal form, i.e., $\phi = \psi_1 \vee \ldots \vee \psi_n$ and each formula $\psi_i$ is a
    conjunction of literals.
14: for all $\psi_i$ in $\phi$ do
15:    Replace $\psi_i$ in $\phi$ by a disjunction of all ground instances of $\psi_i$.
16: end for
```

The algorithm starts with $\phi = \texttt{next}(f')$. Then, it selects a positive literal l in ϕ and unrolls this literal, that is, it replaces l with the bodies of all rules $h\,:\text{-}\,b \in D$ whose head h is unifiable with l with a most general unifier σ (lines 9, 10). The replacement is repeated until all predicates that are left are either of the form $\texttt{true}(t)$, $\texttt{distinct}(t_1, t_2)$ or recursively defined. Recursively defined predicates are not unrolled to ensure termination of the algorithm. Finally, we transform ϕ into disjunctive normal form and replace each disjunct ψ_i of ϕ by a disjunction of all of its ground instances in order to get a ground formula ϕ.

Note that in line 4, we replace every occurrence of **`does`** with **`legal`** to also include the preconditions of the actions that are executed in ϕ. As a consequence the resulting formula ϕ is not equivalent to $\texttt{next}(f')$. However, $\texttt{next}(f') \supset \phi$, under the assumption that only legal moves can be executed, i.e., $\texttt{does}(r, a) \supset \texttt{legal}(r, a)$. This is sufficient for constructing a fluent graph from ϕ.

Note, that we do not select negative literals for unrolling because the construction of our fluent graph only requires positive preconditions for fluents. Still, the algorithm could be easily adapted to also unroll negative literals at the cost of an increased size of the created ϕ.

4.2 Selecting Preconditions for the Fluent Graph

If there are several literals of the form $\texttt{true}(f)$ in a disjunct ψ of the formula ϕ constructed above, we have to select one of them as source of the edge in the fluent graph. As already mentioned, the distance $\Delta(s, f)$ computed with the help of the fluent graph is a lower bound on the actual number of steps needed. To obtain a good lower bound, that is, one as large as possible, the paths between nodes in the fluent graph should be as long as possible. Selecting the best fluent graph, i.e., the one which maximizes the distances, is impossible since this depends on the states we encounter when playing the game, and we do not know these states beforehand. In order to generate a fluent graph that provides good distance estimates, we use several heuristics when we select literals from disjuncts in the DNF of $\texttt{next}(f')$:

First, we only add new edges if necessary. That means, whenever there is a literal $\texttt{true}(f)$ in a disjunct ψ such that the edge (f, f') already exists in the fluent graph, we select this literal $\texttt{true}(f)$. The rationale of this heuristic is that paths in the fluent graph are longer on average if there are fewer connections between the nodes.

Second, we prefer a literal $\texttt{true}(f)$ over $\texttt{true}(g)$ if f is more similar to f' than g is to f', that is $sim(f, f') > sim(g, f')$.

We define the similarity $sim(t, t')$ recursively over ground terms t, t':

$$sim(t,t') \overset{\mathrm{def}}{=} \begin{cases} 1 & t, t' \text{ have arity 0 and } t = t' \\ \sum_i sim(t_i, t'_i) & t = f(t_1, \ldots, t_n) \text{ and} \\ & t' = f(t'_1, \ldots, t'_n) \\ 0 & else \end{cases}$$

In human made game descriptions, similar fluents typically have strong connections. For example, in Tic-Tac-Toe `cell(3,1,x)` is more related to `cell(3,1,b)` than to `cell(b,3,x)`. By using similar fluents when adding new edges to the fluent graph, we have a better chance of finding the same fluent again in a different disjunct of ϕ. Thus we maximize the chance of reusing edges.

5 Applying Distance Features

For using the distance function in our evaluation function, we define the normalized distance $\delta(s, f)$.

$$\delta(s,f) \overset{\mathrm{def}}{=} \frac{\Delta(s,f)}{\Delta_{max}(f)}$$

The value $\Delta_{max}(f)$ represents the longest *finite* distance $\Delta_G(g, f)$ from any fluent g to f in G.

Thus the normalized distance $\delta(s, f)$ will be infinite if and only if $\Delta(s, f) = \infty$, i.e., there is no path from any fluent in s to f in the fluent graph. In all other cases it holds that $0 \leq \delta(s, f) \leq 1$.

Note, that the construction of the fluent graph and computing the shortest paths between all fluents, i.e., the distance function Δ_G, need only be done once for a game. Thus, while construction of the fluent graph is more expensive for complex games, the cost of computing the distance feature $\delta(s, f)$ (or $\Delta(s, f)$) only depends (linearly) on the size of the state s.

5.1 Using Distance Features in an Evaluation Function

To demonstrate the application of the distance measure presented, we use a simplified version of the evaluation function of Fluxplayer [9] implemented in Prolog. It takes the ground DNF of the goal rules as first argument, the current state as second argument and returns the fuzzy evaluation of the DNF on that state as a result.

```
1 eval((D1; ...; Dn), S, R) :- !,
2   eval(D1, S, R1), ..., eval(Dn, S, Rn),
3   R is sum(R1, ..., Rn) - product(R1, ..., Rn).
4 eval((C1, ..., Cn), S, R) :- !,
5   eval(C1, S, R1), ..., eval(Cn, S, Rn),
6   R is product(R1, ..., Rn).
7 eval(not(P), S, R) :- !, eval(P, S, Rp), R is 1 - Rp.
8 eval(true(F), S, 0.9) :- occurs(F, S),!.
9 eval(true(F), S, 0.1).
```

Disjunctions are transformed to probabilistic sums, conjunctions to products, and **true** statements are evaluated to values in the interval $[0, 1]$, basically resembling a recursive fuzzy logic evaluation using the product t-norm and the corresponding probabilistic sum t-conorm. The state value increases with each conjunct and disjunct fulfilled.

We compare the evaluation to a second function that employs our relative distance measure, encoded as predicate `delta`. We obtain the distance-based evaluation function by substituting line 9 of the previous program by the following:

```
eval(true(F), S, R) :- delta(S, F, Dist), Dist =< 1, !,
  R is 0.8*(1-Dist) + 0.1.
eval(true(F), S, 0).
```

Here, we evaluate a fluent that does not occur in the current state to a value in $[0.1, 0.9]$ and, in case the relative distance is infinite, to 0 since this means that the fluent cannot hold anymore.

5.2 Tic-Tac-Toe

Although on first sight Tic-Tac-Toe contains no relevant distance information, we can still take advantage of our distance function. Consider the two states as shown in Figure 2. In state s_1 the first row consists of two cells marked with an `x` and a blank cell.

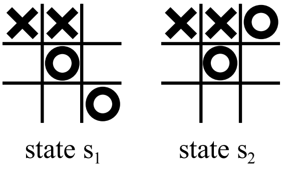


Fig. 2. Two states of Tic-Tac-Toe. The first row is still open in state s_1 but blocked in state s_2.

In state s_2 the first row contains two `x`s and one cell marked with an `o`. State s_1 has a higher state value than s_2 for `xplayer` since in s_1 `xplayer` has a threat of completing a line in contrast to s_2. The corresponding goal condition for `xplayer` completing the first row is:

```
line(x) :- true(cell(1,1,x)),
  true(cell(2,1,x)), true(cell(3,1,x)).
```

When evaluating the body of this condition using the above fuzzy evaluation, we see that it cannot distinguish between s_1 and s_2 because both have two markers in place and one missing for completing the line for `xplayer`, resulting in an evaluation $\mathtt{R} = 1 * 1 * 0.1 = 0.1$

However, the distance-based function evaluates **`true`**`(cell(3,1,b))` of s_1 to 0.1 and **`true`**`(cell(3,1,o))` of s_2 to 0. Therefore, it can distinguish between both states, returning $\mathtt{R} = 0.1$ for $\mathtt{S} = s_1$ and $\mathtt{R} = 0$ for $\mathtt{S} = s_2$.

5.3 Breakthrough

The second game is Breakthrough, again a two-player game played on a chessboard. Like in chess, the first two ranks contain only white pieces and the last two only black pieces. The pieces of the game are only pawns that move and capture in the same way as pawns in chess, but without the initial double advance. Whoever reaches the opposite side of the board first wins. [2] Figure 3 shows the initial position for Breakthrough. The arrows indicate the possible moves, a pawn can make.

The goal condition for the player `black` states that black wins if there is a cell with the coordinates `X,1` and the content `black`, such that `X` is an index (a number from 1 to 8 according to the rules of `index`):

```
goal(black, 100) :- index(X),true(cellholds(X, 1, black)).
```

Grounding this rule yields

```
goal(black, 100) :- true(cellholds(1, 1, black) ;
  ...; true(cellholds(8, 1, black)).
```

We omitted the `index` predicate since it is true for all 8 ground instances.

[2] The complete rules for Breakthrough as well as Tic-Tac-Toe can be found under `http://ggpserver.general-game-playing.de/`

The standard evaluation function cannot distinguish any of the states in which the goal is not reached because **true**(cellholds(X, 1, black)) is false in all of these states for any instance of X.

The distance-based evaluation function is able to construct a fluent graph as depicted in Figure 4 for distance calculation.

Therefore evaluations of atoms of the form **true**(cellholds(X, Y, black)) have now 9 possible values (for distances 0 to 7 and ∞) instead of 2 (true and false). Hence, states where black pawns are nearer to one of the cells (1,8), ..., (8,8) are preferred.

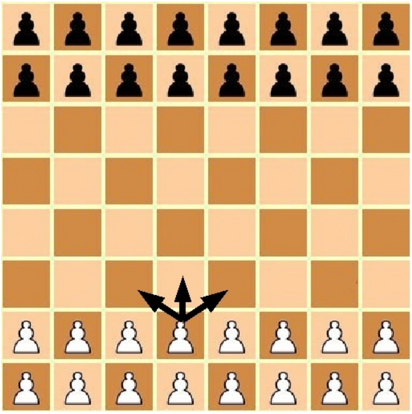


Fig. 3. Initial position in Breakthrough and the move options of a pawn

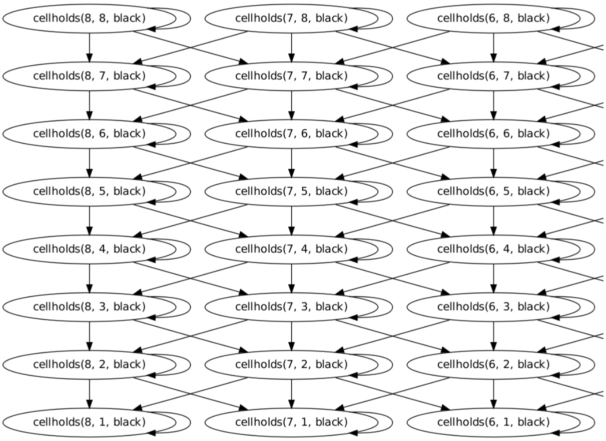


Fig. 4. A partial fluent graph for Breakthrough, role black

Moreover, the fluent graph, and thus the distance function, reflect what could be called "strategic positioning": states with pawns on the side of the board are worth less than those with pawns in the center. This is due to the fact that a pawn in the center may reach more of the 8 possible destinations than a pawn on the side.

6 Evaluation

We first measure the generality and time requirements of our approach for 201 games available at http://ggpserver.general-game-playing.de/.

Figure 5 shows the number of games grouped by the minimum time limit (in seconds) required for successful fluent graph construction.

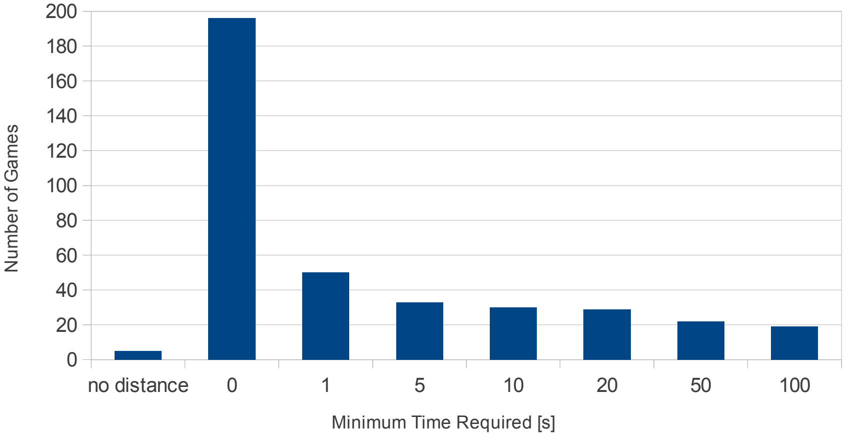


Fig. 5. Number of Games grouped by the minimum time limit required for construction of the fluent graph

We can see that the approach is able to find distance features in all but 5 games. Construction is typically fast and takes a few seconds for the majority of games. There are, however, 33 games that require at least 5 seconds for fluent graph construction and 22 of these even more than 50 seconds. Our interpretation of the results is that the approach is general, although time constraints may pose a problem in some cases and should be addressed, e.g., using time-outs.

For evaluation of the playing strength we implemented our distance function and equipped the agent system Fluxplayer [9] with it. We then set up this version of Fluxplayer ("flux_distance") against its version without the new distance function ("flux_basic"). We used the version of Fluxplayer that came in 4th in the 2010 championship. Since flux_basic is already endowed with a distance heuristic, the evaluation is comparable to a competition setting of two competing heuristics using distance features.

We chose 19 games for comparison in which we conducted 100 matches on average. Figure 6 shows the results.

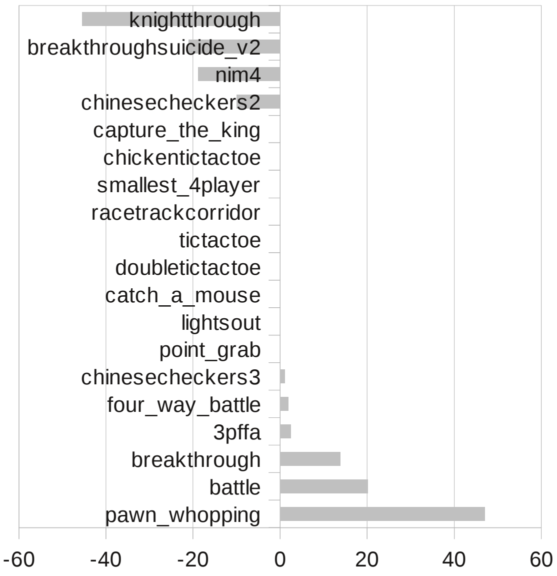


Fig. 6. Advantage in Win Rate of flux_distance

The values indicate the difference in win rate, e.g., a value of $+10$ indicates that flux_distance won 55% of the games against flux_basic winning 45%. Obviously the proposed heuristics produces results comparable to the flux_basic heuristics, with both having advantages in some games. This has several reasons: Most importantly, our proposed heuristic, in the way it is implemented now, is more expensive than the distance estimation used in flux_basic. Therefore the evaluation of a state takes longer and the search tree can not be explored as deeply as with cheaper heuristics. This accounts for three of the four underperforming games. For example in nim4, the flux_basic distance estimation provides essentially the same results as our new approach, just much cheaper. In chinesecheckers2 and knightthrough, the new distance function slows down the search more than its better accuracy can compensate.

On the other hand, flux_distance performs better in complicated games. There the higher accuracy of the heuristics typically outweighs the disadvantage of the heuristics being slower.

Interestingly the higher accuracy of the new distance heuristics is the reason for flux_distance losing in breakthrough_suicide. The game is exactly the same as breakthrough, however, the player to reach the other side of the board first does not win but loses.

The heuristics of both flux_basic and flux_distance are not good for this game since both are based the minimum number of moves necessary to reach the goal while the optimal heuristic would depend on the maximum number of moves available to avoid losing. However, since flux_distance is more accurate, flux_distance selects even worse moves that flux_basic. Specifically, flux distance tries to maximize (a much more accurate) minimal distance to the other side of the board, thereby allowing the opponent to capture its advanced pawns. This behavior, however, results in a smaller maximal number of moves until the game ends and forces to advance the few remaining pawns quickly. Thus, the problem in this game is not the distance estimate but the fact that the heuristic is not suitable for the game.

Finally, in some of the games no changes were found since both distance estimates performed equally well. However, rather specific heuristics and analysis methods of flux_basic could be replaced by our new general approach. For example, the original Fluxplayer contains a special method to detect when a fluent is unreachable, while this information is automatically included in our distance estimate.

7 Related Work

Distance features are part of classical agent programming for games like chess and checkers in order to measure, e.g., the distance of a pawn to the promotion rank. A more general detection mechanism was first employed in Metagamer [8] where the features "promote-distance" and "arrival-distance" represented a value indirectly proportional to the distance of a piece to its arrival or promotion square. However, due to the restriction on symmetric chess-like games, the features are still too specific for an application in GGP.

Currently, a number of GGP agent systems apply distance features in different forms. UTexas [6] identifies order relations syntactically and tries to find 2d-boards with coordinates ordered by these relations. Properties of the content of these cells, such as minimal/maximal x- and y-coordinates or pair-wise Manhattan distances are then assumed as candidate features and may be used in the evaluation function. Fluxplayer [9] generalizes the detection mechanism using semantic properties of order relations and extends board recognition to arbitrarily defined n-dimensional boards.

Another approach is pursued by Cluneplayer [2] who tries to impose a symbol distance interpretation on expressions found in the game description. Symbol distances, however, are again calculated using Manhattan distances on ordered arguments of board-like fluents, eventually resulting in a similar distance estimate as UTexas and Fluxplayer.

Although not explained in detail, Ogre [4] also employs two features that measure the distance from the initial position and the distance to a target position. Again, Ogre relies on syntactic detection of order relations and seems to employ a board centered metrics, ignoring the piece type.

All of these approaches rely on the identification of certain fixed structures in the game (such as game boards) but can not be used for fluents that do not belong to such a structure. Furthermore, they make assumptions about the distances on these structures (usually Manhattan distance) that are not necessarily connected to the game dynamics, e.g., how different pieces move on a board.

In domain independent planning, distance heuristics are used successfully, e.g., in HSP [1] and FF [3]. The heuristics $h(s)$ used in these systems is an approximation of the plan length of a solution in a relaxed problem, where negative effects of actions are ignored. This heuristics is known as delete list relaxation. While on first glance this may seems very similar to our approach, several differences exist:

- The underlying languages, GDL for general game playing and PDDL for planning, are different. A translation of GDL to PDDL is expensive in many games [5]. Thus, directly applying planning systems is not often not feasible.
- The delete list relaxation considers all (positive) preconditions of a fluent, while we only use one precondition. This enables us to precompute the distance between the fluents of a game.
- While goal conditions of most planning problems are simple conjunctions, goals in the general games can be very complex (e.g., checkmate in chess). Additionally, the plan length is usually not a good heuristics, given that only the own actions and not those of the opponents can be controlled. Thus, distance estimates in GGP are usually not used as the only heuristics but only as a feature in a more complex evaluation function. As a consequence, computing distance features must be relatively cheap.
- Computing the plan length of the relaxed planning problem is NP-hard, and even the approximations used in HSP or FF that are not NP-hard require to search the state space of the relaxed problem. On the other hand, computing distance estimates with our solution is relatively cheap. The distances $\Delta_G(f, g)$ between all fluents f and g in the fluent graph can be precomputed once for a game. Then, computing the distance $\Delta(s, f')$ (see Definition 3) is linear in the size of the state s, i.e., linear in the number of fluents in the state.

8 Future Work

One problem of the approach is its computational cost for constructing the fluent graph, that may prevent an application of our distance features for narrow time constraints.

One way to reduce the time needed for construction is a reduction of the size of ϕ via a more selective expansion of predicates (line 5) in Algorithm 1. Developing heuristics for this step is one of the goals for future research.

In addition, we are working on a way to construct fluent graphs from non-ground representations of the preconditions of a fluent to skip the grounding step. For example, the partial fluent graph in Figure 1(a) is identical to the fluent graphs for the other 8 cells of the Tic-Tac-Toe board. The fluent graphs for all 9 cells are obtained from the same rules for **`next`**`(cell(X,Y,_)`, just with different instances of the variables `X` and `Y`. By not instantiating `X` and `Y`, the generated DNF is exponentially smaller while still containing the same information.

The quality of the distance estimates depends mainly on the selection of preconditions. At the moment, the heuristics we use for this selection are intuitive but have no thorough theoretic or empiric foundation. In future, we want to investigate how these heuristics can be improved.

Furthermore, we intend to enhance the approach to use fluent graphs for generalizations of other types of features, such as, piece mobility and strategic positions.

9 Summary

We have presented a general method of deriving distance estimates in General Game Playing. To obtain such a distance estimate, we introduced fluent graphs, proposed an algorithm to construct them from the game rules and demonstrated the transformation from fluent graph distance to a distance feature.

Unlike previous distance estimations, our approach does not rely on syntactic patterns or internal simulations. Moreover, it preserves piece-dependent move patterns and produces an admissible distance heuristic.

We showed on an example how these distance features can be used in a state evaluation function. We gave two examples on how distance estimates can improve the state evaluation and evaluated our distance against Fluxplayer in its most recent version.

Certain shortcomings should be addressed to improve the efficiency of fluent graph construction and the quality of the obtained distance function. Despite these shortcomings, we found that a state evaluation function using the new distance estimates can compete with a state-of-the-art system.

References

1. Bonet, B., Geffner, H.: Planning as heuristic search. Artificial Intelligence 129(1-2), 5–33 (2001)
2. Clune, J.: Heuristic evaluation functions for general game playing. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 1134–1139. AAAI Press (2007)
3. Hoffmann, J., Nebel, B.: The FF planning system: Fast plan generation through heuristic search. JAIR 14, 253–302 (2001)
4. Kaiser, D.M.: Automatic feature extraction for autonomous general game playing agents. In: Proceedings of the Sixth Intl. Joint Conf. on Autonomous Agents and Multiagent Systems (2007)
5. Kissmann, P., Edelkamp, S.: Instantiating General Games Using Prolog or Dependency Graphs. In: Dillmann, R., Beyerer, J., Hanebeck, U.D., Schultz, T. (eds.) KI 2010. LNCS, vol. 6359, pp. 255–262. Springer, Heidelberg (2010)
6. Kuhlmann, G., Dresner, K., Stone, P.: Automatic Heuristic Construction in a Complete General Game Player. In: Proceedings of the Twenty-First National Conference on Artificial Intelligence, pp. 1457–1462. AAAI Press, Boston (2006)
7. Love, N., Hinrichs, T., Haley, D., Schkufza, E., Genesereth, M.: General game playing: Game description language specification. Tech. Rep., Stanford University (March 4, 2008), the most recent version should be available at `http://games.stanford.edu/`
8. Pell, B.: Strategy generation and evaluation for meta-game playing. Ph.D. thesis, University of Cambridge (1993)
9. Schiffel, S., Thielscher, M.: Fluxplayer: A successful general game player. In: Proceedings of the National Conference on Artificial Intelligence, pp. 1191–1196. AAAI Press, Vancouver (2007)
10. Schiffel, S., Thielscher, M.: Automated theorem proving for general game playing. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) (2009)
11. Schiffel, S., Thielscher, M.: A Multiagent Semantics for the Game Description Language. In: Filipe, J., Fred, A., Sharp, B. (eds.) ICAART 2009. CCIS, vol. 67, pp. 44–55. Springer, Heidelberg (2010)

Retrieving Topological Information for Mobile Robots Provided with Grid Maps

David Portugal and Rui P. Rocha

Institute of Systems and Robotics, Department of Electrical and Computer Engineering
University of Coimbra, 3030-290 Coimbra, Portugal
{davidbsp,rprocha}@isr.uc.pt

Abstract. In the context of mobile robotics, it is crucial for the robot to have a consistent representation of the surrounding area. However, common grid maps used in robotics do not provide any evidence as to connectivity, making it harder to find appropriate paths to particular points on the site. Therefore, abstracting the environment where mobile robots carry out some mission can be of a great benefit.

Topological maps have been increasingly used in robotics, because they are fairly simple and an extremely intuitive representation for tasks that involve path planning. In this article, a method for retrieving a topological map from an *a priori* generic grid map of the environment is presented. Beyond extracting a 2D diagram which portrays the topology of the infra-structure, the focus is placed on obtaining graph-like data related to the connectivity of important points in the area, that can be passed on to robots or to a centralized planner, in order to assist the navigation task. The proposed method is further elaborated in detail and its results prove the simplicity, accuracy and efficiency of the approach.

Keywords: Robot navigation, Graphs, Topological maps, Voronoi diagrams.

1 Introduction

In robotics, it is extremely important for autonomous mobile robots to learn and preserve models of the environment. Without an internal description of the environment and information of their position and orientation with respect to this map, most mobile robotic tasks, like driving while avoiding collisions and navigating, would become impractical.

In navigation tasks, it is normally assumed that the environment is known *a priori*. On the other hand, exploration is generally related to completely or partially unknown environments. For both cases, maintaining or building internal representations of the infra-structure is one of the key issues to the successful completion of the task.

The two major distinct paradigms for mapping indoor environments are grid-based maps and topological maps [1]. In the present work, we focus on the latter one.

Grid-based methods produce accurate metric maps, are easy to build, represent and maintain, but often suffer from memory space and time complexity, because of its fine resolution restrictions which results in memory problems and also harder efficient planning and navigating in large-scale infrastructures. Topological maps, on the other hand,

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 204–217, 2013.

produce graph-like maps that can be used much more efficiently. They are simpler, permit efficient planning and do not require accurate determination of the robot's position. In these maps, vertices correspond to important places or landmarks, which are connected by edges that represent paths between them. However, its inaccuracy makes it harder to maintain consistency in large scale environments, which results in difficulties in recognizing similar places that look alike. There are some other constraints related to navigation using topological maps. Some limitations previously identified [2] were: handling inaccurate position and orientation information and detecting neighbor vertices in the topological map by traversing them, as opposed to sensing or recognizing them. Despite these constraints, the path planning method that was used produced adequate results.

In addition, map validation and self-localization in topological maps is also an important issue for correct robot navigation, which has been extensively studied [3]. In the work, the robot is given an input graph-like map and its current position and orientation with respect to the map. The robot has no instrument for measuring distances or orientations. By exploring the world systematically and using distinct markers, which can be detected, dropped and picked up, the robot can recognize places and determine whether the map is correct. The same authors later proposed [4] a multi-robot topological mapping technique. Robots simultaneously explore different regions of the environment and they meet periodically to merge their individual maps in order to create a shared merged map of the complete environment.

As seen before, topological maps can be a very important tool for most tasks using mobile robots. Many works exploit this simple representation to employ correct robot navigation. In Ferreira *et al.* [5], a robot drives around the environment and self-localizes, while using a place recognition technique to build 3D point cloud sets for monitoring changes that might take place in the environment.

In this work, the focus is mainly on obtaining a global topological abstraction from a preexisting metric representation of an environment. For example, for surveillance, monitoring and patrolling tasks with multiple robots, it is common to rely on topological maps for navigation issues. In a previous work [6], we presented a novel multi-robot patrolling algorithm based on topological maps.

In this article, we describe in detail how complete topological information is extracted from an existing 2D grid map representation of the area to be patrolled, which in turn can be obtained with a state-of-the-art robotic mapping technique, *e.g.* [7].

The next section presents a survey of previous techniques for extracting topological representations like diagrams and graphs from metric representations given *a priori*. Typically, these metric representations are occupancy grids, which are probabilistic maps wherein each cell of the grid contains a probability value that indicates whether the related location is free space or part of an obstacle [8]. These grids are usually obtained in a preceding exploration phase. In Section 3, we state the problem to be addressed and Section 4 presents the algorithm proposed to solve it. Later on, results are presented and the article ends with conclusions and future work.

2 Related Work

In the literature, there are a few existing techniques to extract topological representations using metric maps. In particular, Voronoi Diagrams [9] have been extensively used to plan a path that stays away from obstacles as far as possible. In addition, the Generalized Voronoi Diagram (GVD) [10] was described as a “retraction of the free space of the working space onto a network of one-dimensional curves reflecting the connectivity of free space” and a hierarchically organized Voronoi-based route graph representation for robot navigation in indoor scenarios was proposed.

Furthermore, the Extended Voronoi Graph (EVG) [11] has been presented more recently. In this new approach, the diagram (or skeleton) computation overcomes limitations in the sensory horizon of robots by clearly defining a transition from corridor-following to wall-following in large rooms. Although defined as a graph, beyond the actual diagram no information about vertices or edges is available after the final computation.

Additionally, Kolling and Carpin [12] proposed a method to extract graph representations from occupancy grids for surveillance tasks, based on the GVD. Graph modifications, called contractors, are promoted to simplify the surveillance problem towards solving it using fewer robots.

Another commonly used technique when extracting graphs is the Delaunay triangulation (DT) [13]. In fact, the DT of a discrete point set generally corresponds to the dual graph of the Voronoi diagram for the same point set.

Katsilieris *et al.* [14] extract the traversability graph from the original metric map in the context of searching intruders with a team of robots. Firstly, the obstacle-free area is triangulated and then triangles are merged to form large convex regions where the vertices are extracted. The set of edges consists of all pairs of regions that share a side.

In addition, a technique called Reduced Constrained Delaunay Triangulation to build graphs is also becoming popular. This technique is based on a DT computation together with a mechanism to reduce the graph by minimizing the distance each robot has to traverse. It was applied by Fazli *et al.* [15] in the context of a multi-robot area coverage task in a known and static 2D environment.

Another method presented in the literature to extract the graph of an environment is the visibility graph [16], which is composed of straight lines joining a sequence of vertices that are normally placed near the limits of obstacles in the environment (considering the dimension and pose of the robot). The graph is created connecting all vertices that can “see” each other, hence the name visibility graph.

Among the most used techniques are also thinning [17] and skeletonization [18] methods. These are operations used to remove the foreground pixels from binary images, analogously to peeling an onion. They provide a skeleton of the image, reducing all lines to a single pixel thickness. The output of such techniques is similar to the GVD. Szabó [18] also refers in his work some other methods for extracting topological maps from occupancy grids, namely sparse pixel approaches and matching opposite contours. Moreover, another technique called C-cells can be found in the literature [19]. It has the principle to divide the space not occupied by obstacles into convex polygons and the vertices of the graph are positioned in the center of those polygons, having each vertex connected in a straight line to the closest visible vertex.

Most previous works mentioned above present methods to visually extract the topological representation. However, many of these do not correctly characterize vertices and edges on the graph (like Voronoi Diagrams, EVG, thinning or skeletonization). As for those that do, the methods which are based on Delaunay Triangulation need a manual definition of the vertices in the environment to compute the topology and usually generate non-elegant and unbalanced representations. In the case of visibility graphs, highly inadequate topologies are produced since the vertices are positioned very close to obstacles, which is unnatural for robot navigation. The Generalized Voronoi Graph (GVG), which consists of a vertex for every meet and end point of the GVD and edges reflecting the connectivity of the GVD, results in very complex graphs with a large number of vertices and edges that are mostly redundant, especially when considering navigation tasks for robots. In addition, all of these methods assume that the edges connecting pairs of neighboring vertices must be a straight line, which is not necessarily true for robots with the ability to locally plan paths between two vertices.

In this work, we intend to go further ahead, by firstly extracting an elegant topological representation and then focusing on identifying vertices, edges and their length (*i.e.*, their weight) and obtaining relevant data concerning the vertices and the connectivity between areas of the map so as to assist robots navigation in tasks like patrolling, coverage, monitoring, surveillance, pursuit-evasion and others.

3 Problem Statement

In this work, the environment is assumed to be known *a priori* and a metric representation, typically with the form of an occupancy grid, is available. The goal is to abstract the environment through a topological representation, *i.e.*, a graph, and obtain all information about the graph's connectivity.

The techniques described in the previous section will lead to different graph representations, like placing vertices close to obstacles, inserting them halfway between obstacles or positioning them in a uniform way along the terrain. Having this in mind, a careful choice should be made, considering the applications in which the extraction is applied.

We start with a skeleton representation and then focus on extracting topological information to characterize the connectivity of the environment. There are several techniques to compute the initial skeleton as seen previously in section 2. In this work, it is obtained via the EVG computation, mainly because of its results in terms of producing visually attractive path representations that stay away from obstacles and that consider the sensing range of the robots as an input of the method. Additionally, it is a relatively recent technique, which is open source and is easily adaptable to the code that was developed. Having the representation of the skeleton, the graph is obtained through image processing techniques by correct identification of vertices and edges. The topological map is modeled as an undirected graph. Vertices represent places and edges represent the connectivity (in both directions) between those places.

In the next section, we present the algorithm to extract the complete topological information and its simplicity and efficiency becomes clear.

Table 1. EVG-THIN parameters [11]

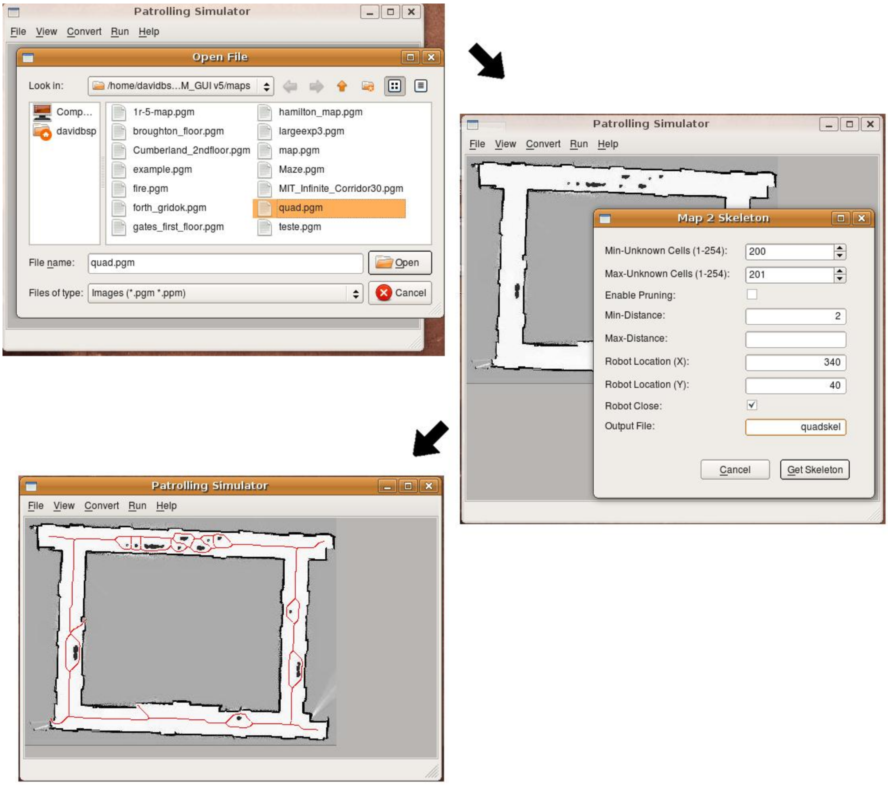
Command	**Description**
min-unknown [N]	The minimum greyscale value (1-254) of unknown cells. Occupied cells are 0-(N-1).
max-unknown [M]	The maximum greyscale value (1-254) of unknown cells. Free cells are (M+1)-255.
pruning [0|1]	Turns pruning on or off. Pruning removes all “branches” of the skeleton except those that meet one of these conditions: 1) The branch touches the edge of the grid. 2) The branch touches unknown cells.
min-distance [R]	Bleeds obstacles by R cells before calculating skeleton. This removes branches that come too close to obstacles.
max-distance [S]	If the skeleton exceeds the S cells from the nearest occupied cells, it switches to following the occupied cells S away.
robot loc [X Y]	This location is used to select which skeleton is valid, given complex images with multiple, disjoint skeletons. By default, the “robot” is located at the centre of the image.
robot-close [0|1]	The robot location (see above) is used to choose which skeleton is valid (if multiple exist). This is done by Euclidean distance between the robot’s location at the skeletal points. This option turns off the checking mechanism except for points where the robot’s distance is within the distance of the skeletal point to its closest obstacle.

4 The Algorithm

In a previous work [6] a patrolling simulator was used to validate a multi-robot patrolling approach based on balanced graph partitioning. Here, we show how the preassumed graph is provided to the patrolling simulator. The algorithm is comprised of 4 steps which are detailed below.

4.1 Acquiring the Skeleton

A tool named EVG-THIN developed by Patrick Beeson [11] was used to acquire skeletons from occupancy grids in this project. “EVG” stands for the Extended Voronoi Graph, which although being called a graph, does not present output related to information about vertices and edges on the graph, giving instead an enhanced skeleton on top of a grid map, when compared to the GVD. “THIN” accounts for the pixel-based thinning algorithm that finds skeletons of bitmaps, which is a fast approximation of the Voronoi diagram. Its code was written to be applied in real-time to occupancy grids, where cells are either occupied, free, or unknown, but it also works on any grayscale bitmap image in other domains. This command-line application runs in any Linux console upon the definition of some parameters, such as the pixels’ threshold for considering them free, unknown or occupied; or the robot’s minimum distance to an obstacle to account for the



Look in: /home/davidbs...M_GUI v5/maps

1r-5-map.pgm
broughton_floor.pgm
Cumberland_2ndfloor.pgm
example.pgm
fire.pgm
forth_gridok.pgm
gates_first_floor.pgm
hamilton_map.pgm
largeexp3.pgm
map.pgm
Maze.pgm
MIT_Infinite_Corridor30.pgm
quad.pgm
teste.pgm

File name: quad.pgm

Files of type: Images (*.pgm *.ppm)

Min-Unknown Cells (1-254): 200

Max-Unknown Cells (1-254): 201

Enable Pruning: ☐

Min-Distance: 2

Max-Distance:

Robot Location (X): 340

Robot Location (Y): 40

Robot Close: ☑

Output File: quadskel

Fig. 1. EVG-THIN incorporated in the simulator

robot's geometric dimension and sensing range when navigating in the map, as seen in Table 1. Setting these parameters correctly is important, because the computed skeleton and final topology depends substantially on those parameters.

The Patrolling Simulator interface displays a window to set all the input parameters reusing and evolving the EVG-THIN's code, which is run in background[1], obtaining the 1 pixel-thin skeleton of the environment, as seen on Figure 1 (right).

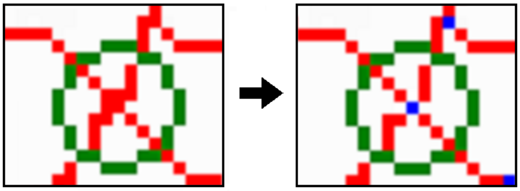


Fig. 2. Cluster detection and correction. The blue dots correspond to vertices subsequently identified.

[1] Note that this program is released under the GNU General Public License (GPL), which is intended to guarantee the freedom to share and change free software.

4.2 Cluster Image Removal

Due to aliasing, sometimes the skeleton representation generated in the previous section presents clusters, *i.e.*, sets of 4 pixel squares, as shown on Figure 2. These clusters are problematic, in the sense that they interfere with the subsequent detection of vertices in the graph, due to the 1 pixel-thin skeleton assumption. Therefore, an intermediate phase to avoid future data corruption when detecting the vertices and edges is necessary: Cluster Removal.

A protective filter was developed to detect and remove such clusters. Basically, it sweeps the image and when a cluster is detected, two actions are possible:

- Pixel Removal: Eliminate unnecessary pixels from the cluster;
- Pixel Shifting: Move pixels to the cluster's neighborhood.

These actions are conducted without affecting the connectivity of the graph. In Figure 3, on the left side, two examples of their application are shown. On top, the 2 pixels marked by a cross are removed, because they are unnecessary to guarantee the connection between all branches.

When a cluster is identified, the filter checks the implications of removing pixels; more specifically it guarantees that no disconnections resulting from erasing pixels will exist, by inspecting all the neighbor pixels around the cluster.

Erasing pixels is not always possible, as it can be observed in the bottom left of Figure 3, where removing red pixels would create gaps in the skeleton underneath. Hence, a pixel shifting strategy was created to remedy such situations.

4.3 Detecting Vertices

Having the cluster-free skeleton, the next step is extracting all vertices of the graph by image processing, which is done on top of the previously generated skeleton. The process is fairly simple, due to the 1-pixel thin assumption. Vertices correspond to single pixels that are placed in two specific locations:

- Dead-ends: Pixels that typically only have one neighbor pixel that also belongs to the skeleton. They originate vertices with degree[2] one.
- Branch intersections: Crucial pixels that correspond to crossings of the branches of the skeleton. They usually have three or more neighbor pixels that also belong to the skeleton, within the eight pixels around them. They originate higher degree vertices.

The process shown in pseudo-code by Algorithm 1 is conducted to identify all vertices in the graph. Basically, the intent is to inspect the 9-pixel window centered on the analyzed pixel of the skeleton and verify if it is a dead-end or a branch intersection. When the vertex is identified, a single blue pixel is drawn in the correct spot. Note that this algorithm relies on the fact that no clusters exist in the graph.

After running the algorithm, the visual representation of the graph on top of the grid is complete, as it will be shown on the results section, where the vertices were highlighted for better identification.

[2] "The degree (or valency) $d_G(v) = d(v)$ of a vertex v is the number $|E(v)|$ of edges at v (...), this is equal to the number of neighbours of v" [20].

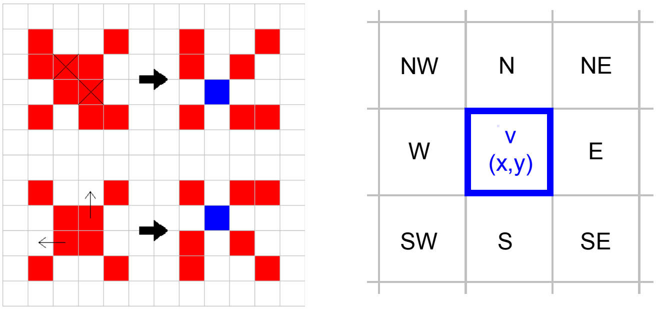
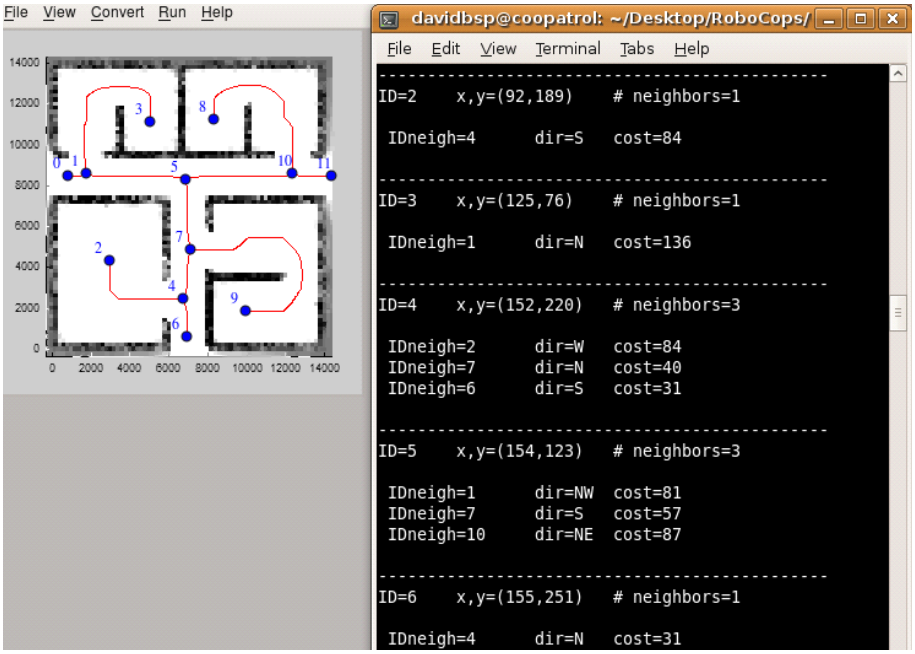


Fig. 3. On the left side: Cluster Analysis - Pixel removal on top and pixel shifting below. On the right side: Possible neighbor directions.


File View Convert Run Help
14000
12000
10000
8000
6000
4000
2000
0
0 2000 4000 6000 8000 10000 12000 14000
0
1
2
3
4
5
6
7
8
9
10
11
davidbsp@coopatrol: ~/Desktop/RoboCops/
File Edit View Terminal Tabs Help

```
---------------------------------------------
ID=2    x,y=(92,189)    # neighbors=1

 IDneigh=4      dir=S   cost=84

---------------------------------------------
ID=3    x,y=(125,76)    # neighbors=1

 IDneigh=1      dir=N   cost=136

---------------------------------------------
ID=4    x,y=(152,220)   # neighbors=3

 IDneigh=2      dir=W   cost=84
 IDneigh=7      dir=N   cost=40
 IDneigh=6      dir=S   cost=31

---------------------------------------------
ID=5    x,y=(154,123)   # neighbors=3

 IDneigh=1      dir=NW  cost=81
 IDneigh=7      dir=S   cost=57
 IDneigh=10     dir=NE  cost=87

---------------------------------------------
ID=6    x,y=(155,251)   # neighbors=1

 IDneigh=4      dir=N   cost=31
```

Fig. 4. Extracting the graph and relevant topological information. Example of a simple tree-like graph with 12 vertices.

4.4 Extracting Relevant Topological Information

The last step of the algorithm is to compute the rest of the topological information to completely characterize the graph (edges, connectivity, etc.). In this work, each vertex data structure will have the following fields associated to it:

- ID
- Coordinates (x, y)
- Degree
- Neighbors (Vertex Degree)={ID, direction, edge weight}

Algorithm 1. Pseudo-Code of the vertex detection process

```
foreach pixel p of the image do
    if p is red then
        count ← number of red pixels inside 9-pixel window centered on p;
    else
        count = 0;
    end
    if count > 3 then    // Exclude pixels with 3 or more neighbors that do not correspond to branch intersections.
        foreach two adjacent red pixels around p do
            count = count −1;
        end
        foreach three adjacent red pixels around p do
            count = count −2;
        end
    else if count == 3 then    // Dead-end (with more than one neighbor pixel)
        if two adjacent red pixels around p then
            count = count +1;
        end
    end
    if count > 3 || count = 2 then
        p is a vertex ⇒ blue pixel in p;
    end
end
```

In this step, the main concern is to convert visual information into graph data that can be used by robots to abstract the environment.

The ID and coordinates (x, y) of each vertex are easily extracted by looking for the blue pixels in the picture and assigning them ID's and saving their (x, y) pixel coordinates.

The other fields are obtained subsequently, also by image processing, by following all branches leaving the vertices via the red pixels' routes. When a blue pixel is found at the end of a route, a neighbor vertex is recognized, therefore the degree of the initial pixel is incremented, and a new entry to the neighbors table of the initial vertex is filled: the ID of the newly-detected neighbor vertex, its direction related to the initial vertex and the edge weight between the 2 vertices.

The ID is obtained by relating the coordinates of the newly-detected vertex to the ID list previously computed. The direction is defined in a similar way to the compass rose designation, as show on the right side of Figure 3, *e.g.*, if the direction of the route leading to the neighbor vertex starts in the pixel above the examined vertex, than the direction is defined as North (N). Also, the edge weight between vertices is defined as the cost of travelling from one vertex to another in pixel units, more specifically, the number of red pixels in the edge between both vertices.

Every vertex is identified by a single blue pixel, as shown previously in Figure 2. After obtaining all the data fields, each vertex will have a table of neighbors with the size of its degree and all the edges and the connectivity between neighbor vertices will be well characterized. This is clear in Figure 4. Having all this information, the framework for agents' navigation in the Patrolling Simulator is finally created.

5 Results and Discussion

The algorithm presented in this article was tested using a large diversity of maps that generated elegant and balanced representations that range from simple topologies to complex ones. An important aspect is that the complexity of the topological maps extracted has no relation to the dimension of the environment (in terms of Cartesian distances). The algorithm attained clear and visually attractive graph representations from metric maps of all sizes and shape, presenting no vertices with self-loops, staying away from obstacles and considering the sensing range of the robots; showing that it scales well in terms of environment, as shown in Figures 5-8. Beyond the accurate results obtained, the algorithm has shown to be computationally efficient. The C++ programming language was used and the Graphical User Interface (GUI) for the Patrolling Simulator was implemented using QT Open Source Edition 4.4.3, which is also based on C++ Language[3]. The system ran on Ubuntu 8.10 Intrepid on an AMD Athlon 64 Processor 3500+, 2.21GHz, with 1GB RAM.

Table 2 illustrates the time taken by the algorithm to extract all information from diverse metric maps, which is dependent not only on the amount of pixels in the free area of each metric map, as well as the complexity of the generated graph. Around 95% of the time values shown is spent computing the skeleton of the environment (first step of the Algorithm), which is the responsibility of EVG-THIN. The three other steps of the approach are computed in a few hundredths of second, which is extremely fast considering the application for which it was designed for.

The image pixels presented in the table do not relate in the same way, for every case, to the free pixels detected in each image. It is also clear that the computation time does not depend exclusively on this. It also depends on the graph's dimension as, for example, the fourth and fifth line of the table highlights.

These results attest to the applicability of the method proposed herein. It is a valuable method for obtaining a topological map of the environment from either an *a priori* known metric map or a metric map being built on the fly.

[3] The Patrolling simulator code is available at:
`http://isr.uc.pt/~davidbsportugal/packages`

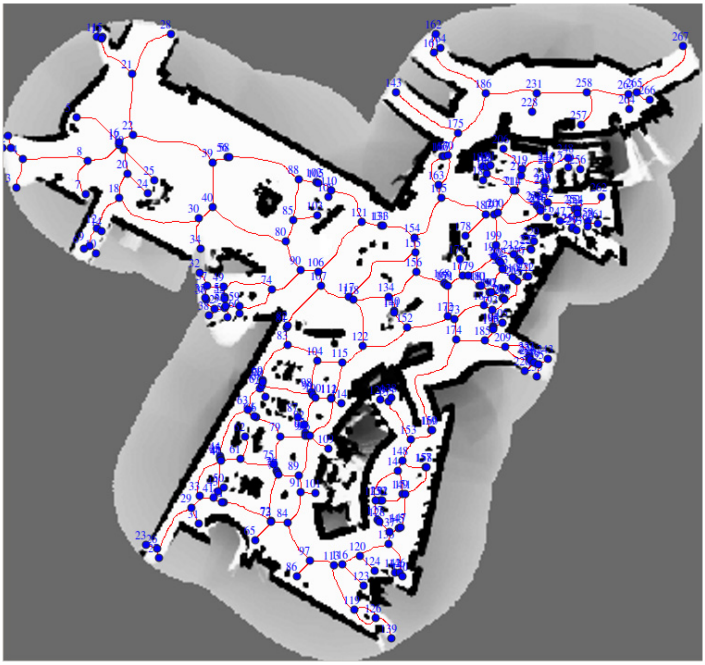


Fig. 5. Results for a complex graph (268 vertices) with high connectivity

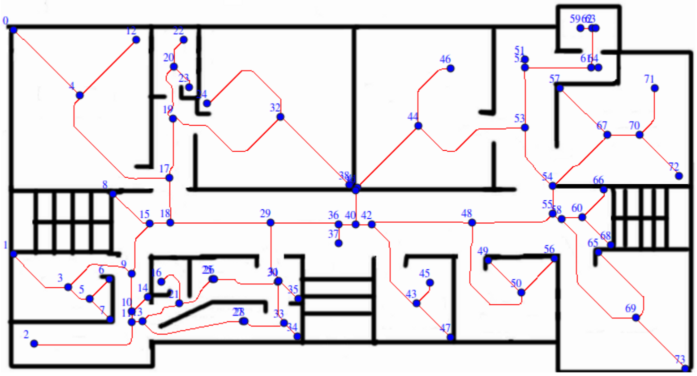


Fig. 6. Results for a medium graph (74 vertices) with defined connectivity

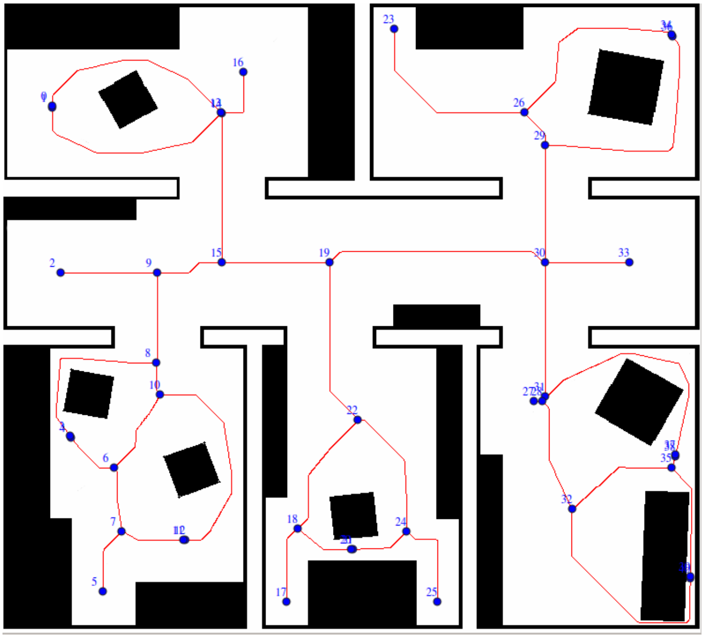


Fig. 7. Results for a medium graph (41 vertices) with defined connectivity

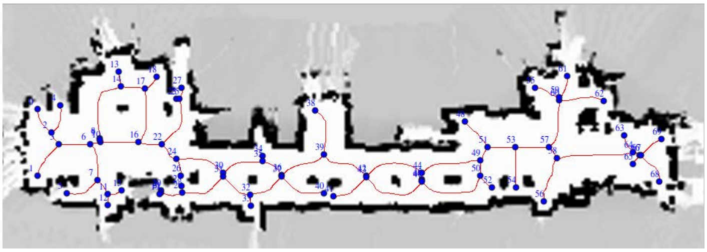


Fig. 8. Results for a graph with 70 vertices and a few connectivity constraints

Table 2. Computation time for different metric maps

Image Pixels	# Vertices	Time (s)
90K	12	0.40
341K	74	1.38
379K	20	1.42
343K	66	1.57
509K	206	1.61
544K	268	1.63
815K	135	3.62

6 Conclusions and Future Work

In this article, a novel way to extract topological maps and connectivity information from standard grid-like grayscale representations is described. The approach presented gains advantage from its simplicity, accuracy and performance. One possible disadvantage of using the EVG-THIN method to compute the skeleton of the grid map is the dependency on the correct parameterization, which is not straightforward for most cases. However, the approach presented in this paper is not limited to the use of EVG-THIN to extract the skeleton, other techniques like those mentioned in Section 2, can also be used.

Unlike most previous works in this area, here the intent is not to present solely a representation of the graph on top of the grid, but also to give one step ahead by proposing a way to convert visual information into data structures, by means of image processing techniques as described.

The proposed approach offers, as output, a complete characterization of the topological aspects of the environment, which has the ability to assist robot's navigation in a broad spectrum of activities, especially those that include path planning. This technique has been recently used to provide a topological map for mobile robots in cooperative patrolling misions [21]

As for future work, it would be interesting to test this approach using different methods in the literature to obtain the underlying diagram to check whether it is possible to speed up the first step of the algorithm without losing quality on the topological representation. Additionally, some questions are still left open like addressing fast update of the Voronoi Diagram given dynamic changes in the environment as well as considering 3D models and deal with topological navigation using mobile robots in real world scenarios.

Acknowledgements. This work was financially supported by a PhD grant (SFRH/BD/64426/2009) from the Portuguese Foundation for Science and Technology (FCT) and the Institute of Systems and Robotics (ISR-Coimbra) also under regular funding by FCT.

References

1. Thrun, S.: Learning maps for indoor mobile robot navigation. Artificial Intelligence 99, 21–71 (1998)
2. Zimmer, U., Fischer, C., Von Puttkamer, E.: Navigation on topologic feature-maps. In: 3rd International Conference on Fuzzy Logic, Neural Nets and Soft Computing, Fukuoka, Japan, pp. 131–132 (1994)
3. Dudek, G., Jenkin, M., Milios, E., Wilkes, D.: Map validation and robot self-location in a graph-like world. Robotics and Autonomous Systems 22(2), 159–178 (1997)
4. Dudek, G., Jenkin, M., Milios, E., Wilkes, D.: Topological exploration with multiple robots. In: 7th International Symposium on Robotics with Applications (ISORA 1998), Anchorage, Alaska, May 10-14 (1998)
5. Ferreira, F., Davim, L., Rocha, R., Dias, J., Santos, V.: Presenting a technique for registering images and range data using a topological representation of a path within an environment. J. of Automation, Mobile Robotics & Intelligent Systems 1(3), 47–55 (2007)

6. Portugal, D., Rocha, R.: Msp algorithm: Multi-robot patrolling based on territory allocation using balanced graph partitioning. In: 25th ACM Symposium on Applied Computing, SAC 2010, Sierre, Switzerland, pp. 1271–1276 (2010)
7. Thrun, S., Bugard, W., Fox, D.: A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In: Int. Conf. on Robotics and Automation, San Francisco, pp. 321–328 (2000)
8. Elfes, A.: Using occupancy grids for mobile robot perception and navigation. Computer 22(6), 46–57 (1989)
9. Voronoi, G.: Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Journal für die Reine und Angewandte Mathematik 134, 198–287 (1908)
10. Wallgrün, J.: Hierarchical voronoi-based route graph representations for planning, spatial reasoning, and communication. In: 4th Int. Cognitive Robotics Workshop (CogRob 2004), pp. 64–69 (2004)
11. Beeson, P., Jong, N., Kuiper, B.: Towards autonomous topological place detection using the extended voronoi graph. In: Int. Conf. on Robotics and Automation (ICRA 2005), Barcelona, pp. 4373–4379 (2005)
12. Kolling, A., Carpin, S.: Extracting surveillance graphs from robot maps. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2008), Nice, France, pp. 2323–2328 (2008)
13. Delaunay, B.: Sur la sphére vide. Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk 7(6), 793–800 (1934)
14. Katsilieris, F., Lindhé, M., Dimarogonas, D., Ögren, P., Johansson, K.: Demonstration of multi-robot search and secure. In: Int. Conf. on Robotics and Automation (ICRA 2010), Anchorage, Alaska, USA (2010)
15. Fazli, P., Davoodi, A., Pasquier, P., Mackworth, A.: Fault-tolerant multi-robot area coverage with limited visibility. In: 2010 IEEE Int. Conf. on Robotics and Automation (ICRA 2010), Anchorage, Alaska, USA (2010)
16. Lozano-Pérez, T., Wesley, M.: An algorithm for planning collision-free paths among polyhedral obstacles. Communications of the ACM 22(10), 560–570 (1979)
17. Ko, B., Song, J., Lee, S.: Realtime building of thinning-based topological map. In: lnt. Conf. on Intelligent Robots and Systems, Sandal, Japan (2004)
18. Szabó, R.: Topological navigation of simulated robots using occupancy grid. Int. J. of Advanced Robotic Systems 1(3), 201–206 (2004)
19. Machado, A.: Patrulha multiagente: Uma análise empírica e sistemática. Master's thesis, Centro de Informática, Universidade Federal de Pernambuco (UFPE), Recife, Brasil (2002)
20. Reinhard, D.: Graph Theory, electronic edition. Springer, Heidelberg (2010)
21. Portugal, D., Rocha, R.P.: On the performance and scalability of multi-robot patrolling algorithms. In: 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR 2011), Kyoto, Japan, November 1-5, pp. 50–55 (2011)

Timeline Planning in the J-TRE Environment

Riccardo De Benedictis and Amedeo Cesta

CNR, Italian National Research Council, ISTC, Rome, Italy
{name.surname}@istc.cnr.it

Abstract. Timeline-based representations constitute a quite natural way to reason on time and resource constraints while planning. Additionally timeline-based planners have been demonstrated as successful in modeling and solving problems in several real world domains. In spite of these successes, any aspect related to search control remains a "black art" for few experts of the particular approach mostly because these architectures are huge application developments environments. For example, the exploration of alternative search techniques is quite hard. This paper proposes a general architecture for timeline-based reasoning that brings together key aspects of such reasoning leaving freedom to specific implementations on both constraint reasoning engines and resolution heuristics. Within such architecture, called J-TRE, three different planners are built and compared with respect to a quite challenging reference problem. The experiments shed some light on key differences and pave the way for future works.

1 Introduction

Common timeline-based planners like EUROPA [1], ASPEN [2], IxTeT [3] and TRF [4,5] are defined as complex software environments suitable for generating planning applications, but quite heavy to foster research work on specific open issues. Some theoretical work on timeline-based planning like [6] was mostly dedicated to explain details of [1] identifying connection with classical planning *a-la* PDDL [7]. The work on IxTeT and TRF have tried to clarify some key underlying principles but mostly succeeded in underscoring the role of time and resource reasoning [8,9]. The search control part has always remained significantly under explored. The current realm is that although these planners capture elements that are very relevant for applications, their theories are often quite challenging from a computational point of view and their performance are rather weak compared with those of state of the art classical planners. Indeed, timeline-based planners are mostly based on the notion of partial order planning [10] and have almost neglected advantages in classical planning triggered from the use of GRAPHPLAN and/or modern heuristic search [11,12]. Furthermore, these architectures rely on a clear distinction between temporal reasoning and other forms of constraint reasoning and there is no sign of attempts to change.

This paper, after a brief introduction on timeline-based planning basics, introduces the J-TRE architecture, an open refinement search framework that encloses common elements of timeline-based planners. Furthermore, three different planners are instantiated which rely on different generic technologies. This approach allows us to fairly compare them as possible alternatives and even propose new ones by opening the way

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 218–233, 2013.

to new possible future evolutions with respect to search control. We also present an interesting comparison of the planner with respect to a quite challenging problem domain.

2 Basics on Timelines

This section introduces some basic concepts. For a more detailed dissertation on timeline based planners the reader should make reference to [13,3,6,4].

2.1 Time, Tokens and Relations: The Plan

To include time into a logic formalism we choose to provide the predicates with extra arguments belonging to the Time domain $\mathbb{T}$ (real or discrete). For example, a predicate $At\,(l)$, denoting the fact that an agent is at a certain location l, can be extended with two temporal arguments $s \in \mathbb{T}$ and $e \in \mathbb{T}$, with $s < e$, representing its starting and ending times, respectively; the $At\,(l, s, e)$ formula would be true only if the agent is at location l from time s to time e. Similarly to what described in [13], we call *token* a proposition that has temporal arguments.

In order to force the proposition arguments to assume the desired values, J-TRE allows the imposition of any kind of linear constraints among the arguments and/or between the arguments and other variables. Since common timeline-based planners typically accept any kind of quantitative temporal interval relations [14] between tokens, that must often be customized by the user, the J-TRE framework facilitate the synthesis of planning domains by allowing the organization of constraints in macros called *relations*.

The task of the solver is to find a legal sequence of tokens that bring the timelines (that constitute the partial and final *plan*) into a final configuration that verifies both the *domain theory*[1] as well as a determined set of desired conditions called *goals*. Starting from an initial state, the planner moves in the search space by adding or removing tokens and/or relations (i.e., changing the current state) until all goals are satisfied.

2.2 Interactions among Tokens: The Timelines

From a planning perspective, the easiest way to describe a *timeline* is to consider it a mere collection of tokens. The predicates that can be accommodated on a timeline as well as the behavior assumed by the planner when a new token is added to a timeline depend on the nature of the timeline itself and, in some cases, on the modeled domain. J-TRE allows the utilization of families of timelines which provide different modeling ability, such as multi-valued state variables [13] as well as renewable and consumable resources like those commonly used in constraint-based scheduling [9].

The *state variable* is the most used type of timeline in this approach to planning. State variable predicates are defined by the user during domain definition. The semantics of a state variable is that for each time instant $t \in \mathbb{T}$ the timeline can assume only one value. This corresponds to a mutual exclusion rule between different tokens. Let us assume,

[1] The set of rules that model the domain's dynamic behavior.

for example, to have a predicate $At\,(l, s, e)$ and a predicate $GoingTo\,(l, s, e)$. We know for sure that tokens assuming At and $GoingTo$ values cannot overlap. However, two tokens both assuming the At value can overlap if and only if their respective parameters (l, s and e) are pairwise constrained to be equal. In this case we talk about *merging* (or, in some cases, *unification*) of tokens.

Now let us suppose that we want a rule stating that every time we are going to a given location we will reach that location. We can enforce such rule by temporally constraining the $GoingTo\,(l, s, e)$ and the $At\,(l, s, e)$ predicates by means of the Allen's *meet* relation having the same location l (see [14]). In other words, for each token t_i with a $GoingTo\,(l, s, e)$ value the planner must ensure that t_i *meets* another token t_j with an $At\,(l, s, e)$ value, either by imposing the *meet* constraint between t_i and t_j if they both exist, or by adding the missing token t_j before enforcing the same constraints. This kind of "rules" are generalized in a concept usually called *compatibility* (again, here we use a terminology consistent with [13]). Compatibilities define causal relations that must be complied to in order for a given token to be valid. Although the syntax can be quite different among various planners, a compatibility can be recursively defined by means of a *reference* predicate and a *requirement* where a requirement can be a slave (or target) predicate, a relation among predicates, a conjunction of requirements or, in rare cases, a disjunction of requirements. It is important to underscore that the compatibilities may often involve predicates defined on different timelines, thus allowing to synchronize concurrent activities on different domain components. Most timeline-based planners admit only conjunctions of requirements and reproduce disjunctions by assigning multiple compatibilities to the same predicate.

To simplify matters, we describe compatibilities through logic implications $reference \rightarrow requirement$. In some cases, we will give tokens a specific name and will address their arguments using a Java style *dot* notation (i.e., given a token t having proposition $P\,(s, e)$ its starting point will be expressed as $t.s$).

Other commonly used types of timelines are the *resources* characterized by a *resource level* $\mathcal{L} : \mathbb{T} \rightarrow \mathbb{R}$, representing the amount of available resource at any given time, and by a *resource capacity* $\mathcal{C} \in \mathbb{R}$, representing the physical limit of the available resource.

We can identify several types of resources depending on how the resource level can be increased or decreased in time. A *consumable resource* is a resource whose level is increased or decreased by some activities in the system. An example of consumable resource is a reservoir which is produced when a plan activity "fills" it (i.e., a tank refueling task) as well as consumed if a plan activity "empties" it (i.e., driving a car uses gas). We model consumable resources through a timeline that has two allowed predicates: a predicate $produce\,(a, s, e)$ to represent a resource production of amount a from time s to time e and a predicate $consume\,(a, s, e)$ to represent a resource consumption of amount a from time s to time e. The planner may need to identify an ordering of the involved activities in order to avoid overproductions (resource level $\mathcal{L}$ cannot exceed capacity $\mathcal{C}$) as well as overconsumption (resource level $\mathcal{L}$ cannot cannot be lower than zero).

The last commonly used timeline type, quite popular in the scheduling literature, is the *reusable resource*. Reusable resources can be modeled as consumable resources that are produced at their start time and are consumed at their end time. We can model reusable resources through a predicate $use\,(a, s, e)$ that is true iff there is a production of resource of amount a at time s and a consumption of resource of amount a at time e. Now let's assume we have two tokens t_0 and t_1 belonging to a reusable resource timeline such that $t_0.s < t_1.e \wedge t_1.s < t_0.e$ (this constraint simply forces their overlapping). The expected behavior of the resource is to have a resource usage of $t_0.a$ during t_0's duration when there isn't overlapping with t_1, a resource usage of $t_0.a + t_1.a$ when t_0 overlaps with t_1, a resource usage of $t_1.a$ during t_1's duration when there is no overlapping with t_0 and a resource usage of 0 elsewhere.

We also want to define constraints on both reusable and consumable resource levels. We are interested in supporting the following constraints making use of special predicates:

- $gt\,(a, s, e)$ to force the profile of the resource to be strictly greater than a
- $ge\,(a, s, e)$ to force the profile of the resource to be greater than a
- $le\,(a, s, e)$ to force the profile of the resource to be lower than a
- $lt\,(a, s, e)$ to force the profile of the resource to be strictly lower than a

3 Reasoning on Timelines

Having defined the basic terminology to describe a partial plan and the timelines, we address the problem of reasoning with such building blocks introducing first our planning architecture. A *Domain Definition Language* called DDL.4 is the entry point to the J-TRE environment, allowing the final user to specify the domain objects of interest, the relevant physical constraints that influence their possible temporal evolutions (e.g., compatibility/coordination constraints among different objects, maximum capacity of resources, etc.), as well as the planning goals.

Common timeline-based planners reach a solution state by applying an iterative refinement procedure. If we call *flaw* every possible inconsistency of the current plan, the role of the planner can be reduced to the identification and the resolution of each flaw in the plan. The planning process proceeds until a consistent plan is found, i.e., the propagation of the solving constraints succeeds and all flaws are eliminated. The general solving strategy broadly entails: (i) identifying a set of flaws, (ii) selecting a flaw according to a *selection strategy*, and (iii) solving it by using a *resolution strategy* (see Figure 1 as a reference). During the solving process, a consistency check routine is called on each domain object, possibly generating new flaws to be solved. The solving procedure ends when a consistent node (i.e., containing no flaws) is found. While flaws can be of different types and can arise for different reasons, what they all have in common is that a *search choice* is necessary to solve each of them. Depending on the reason why a flaw arises, there can basically be four kinds of flaws: (i) *goal flaws*, arising when a new token is added to a state variable to satisfy a compatibility requirement, (ii) *disjunction flaws*, arising when a disjunction statement is found while enforcing some domain rule (expressed in DDL code), (iii) *preference flaws*, arising when a preferred statement is found again while enforcing some domain rule, and (iv) *timeline inconsistency flaws*, arising when an inconsistency is detected on some domain object like, for example, different

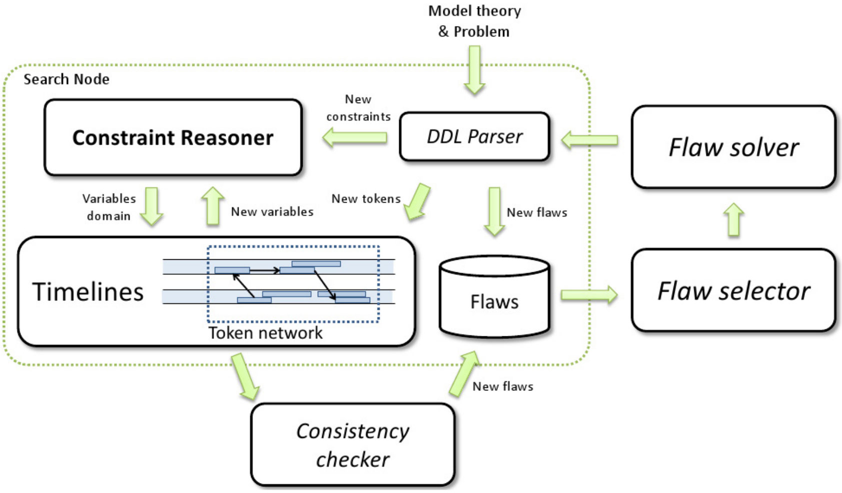


Fig. 1. J-TRE architecture. The planner collects flaws, selects one of them and solves it by executing some DDL code. The planning process will stop in a search space node without flaws.

values overlapping on a state variable, reusable resource oversubscriptions, consumable resources overproductions or overconsumption, etc. Each time a new node of the search space is created or new constraints are added to the current search space node, a *check consistency* routine is called on each object of the domain and, depending on the object itself, further flaws can be added to the current search space node. This procedure is required in order to remove any further inconsistencies from the timelines *scheduling* tokens in time. This technique has been introduced in [4], in which state variables are observed as resources over time and contentions peaks over their continuous representation are removed by adding precedence constraints among tokens.

While, in our system, there is almost no difference in *which* flaw is solved first (as far as we ignore efficiency aspects) because they all have to be solved sooner or later, there could be serious troubles in *how* they are solved, especially in case of cyclic problems.

Consider, for example, a simple state variable having $At(l, s, e)$ and *GoingTo(l,s,e)* as allowed values. Moreover there is a compatibility for predicate At that requires for each token to start at 0 *or* to be met by a $GoingTo$ token with same location. Finally, a compatibility for predicate $GoingTo$ that requires for each token to be met by a predicate At. We have an initial state with a token $At\,(l_0, 0, [1, +\inf])$ and a goal $At\,(l_3, [0, +\inf], [1, +\inf])$. The planner has to apply related compatibility for the goal token producing a sub-goal $GoingTo\,(l_3, [0, +\inf], [1, +\inf])$ than another sub-goal $At\,(l, [0, +\inf], [1, +\inf])$ that can unify with first token or apply another compatibility resulting in another $GoingTo\,(l, [0, +\inf], [1, +\inf])$ possibly leading to an infinite loop planning about the agent going walking around. In short, although scheduling search space, however exponential, is always finite, it can be the case that compatibility application space is infinite.

Although a crafty strategy does not exist yet (exception made for some work by Bernardini [15]) the idea we have pursued is to proceed in depth on the search space maintaining a bound on the number of conflicts. If failing to solve a node within that bound, the overall solving procedure will backtrack to the lowest possible level restarting the search. However, additional solutions to this problem still need to be investigated.

Among the advancements offered by the J-TRE software infrastructure w.r.t. to previous timeline representation frameworks, such as the TRF [5], we underscore the following: (i) the "unification" of the concept of *flaw* (i.e., a plan inconsistency) into a single entity that is uniformly treated (and reasoned upon) throughout the whole J-TRE infrastructure. In J-TRE, flaw analysis and management is no longer spread across specialized reasoners depending on the flaw type, thus allowing to introduce more effective search heuristics that exploit the cross-comparisons among flaws of different types; (ii) the possibility to express constraint of increasing complexity among different domain parameters (e.g., modeling the dependency between resource quantity to be produced and the production activity duration, etc.); (iii) the introduction of the consumable resources among the timeline types.

4 A Timeline-Based Planner Portfolio

Depending on the underlying available technology, the flaw solving procedure can decide if creating a branch on the search tree or adding disjunctions to the underlying constraint reasoner. Information from constraint reasoning can be exploited in flaw selection and flaw resolution strategies.

Several scenarios can arise from having a single search space node with all possible disjunctions to having a huge search space in which each node has its own constraint reasoner.

4.1 Arc Consistency and Timeline-Based Planning: J-TRE *(ac)*

Our first attempt at solving timeline-based planning problem has required the development of a simple *arc consistency* algorithm (namely AC-3 [16]) for solving *Constraint Satisfaction Problems* (CSP) and manage numeric variables (i.e., time-points, production amounts, etc...). The CSP problem is defined as:

- A set of *variables* $X = \{x_1, \ldots, x_n\}$.
- For each variable x_i, a finite set D_i of possible values (its *domain*). For our purposes we consider only numeric variables so we can represent the domain through a simply couple $[lb, ub]$ representing all possible values between lb and ub (bound consistency).
- A set of *constraints* restricting the values that the variables can simultaneously take.

A *solution* to a CSP is an assignment of a value from its domain to every variable, in such a way that all constraints are satisfied at once.

For each variable x_i, the algorithm maintains a *watch list* of constraints "watching" x_i. A queue data structure maintains all variables whose domain has been updated. While the queue is not empty, a variable is selected from the queue and propagation is performed on all constraints of the watch list associated to the variable possibly updating other variables and enqueueing them. The algorithm goes on until the queue becomes empty (the CSP is arc consistent) or a variable domain becomes empty (the CSP is inconsistent).

If we consider our CSP as a directed graph with nodes representing variables of the problem and arcs between variables representing constraints, the worst-case time complexity of AC-3 algorithm is $O\left(e \cdot d^3\right)$ where e is the number of arcs and d is the size of the largest variable domain.

It is worth underscoring that two key complexity factors here are: (i) the need to tackle huge domains (e.g., $[0, +\inf]$ is a common domain for temporal arguments) and (ii) possible presence of cyclic networks. A simple update to the algorithm consists in limiting the number of possible updates of each variable to the number of constraints of the CSP. Exceeding this limit would obviously determine the existence of a cycle that incrementally would empty the domain of some variable involved in the cycle itself resulting in an inconsistent CSP. This fact allowed us to move worst-case time complexity of our AC-3 algorithm to $O\left(e \cdot \min\left(e, d\right)^3\right)$ removing, in most cases, the discouraging domain size from time complexity.

A further required extension to the AC-3 algorithm is to make it backtrackable. A queue data structure maintains, for each level, the domain of the variables before being updated as well as the added constraints. When backtracking, we can restore variable domains and remove added constraints from watch lists. It is worth noting that this architecture allows us to maintain a single CSP reasoner for the entire timeline problem solving procedure. Indeed, not constrained variables have an empty watch list and are not involved in the propagation procedure.

***Flaw Management in J-TRE* (ac).** First thing we do is to generate the initial partial plan by executing the DDL code representing our problem. Each time a state variable token is found in the DDL code, a new goal flaw is added to the set of flaws of the current search space node. In case of a DDL disjunction, a disjunction flaw is added. Finally, in case a preference is found, a new preference flaw is added.

The flaw which is more easy to describe is the preference flaw. Such flaws can be managed simply by adding a branch on the search space and executing the preferred DDL code on one of the nodes. Disjunction flaws can be managed in a similar way by adding a branch on the search space and executing each DDL code snippet on each child node.

For what concerns goal flaws, we have two possible resolution strategies: *compatibility application* and *merging*. The flaw solver will generate a branch on the search space with several nodes representing all possible merges and a node for the compatibility application. Compatibility application is managed simply executing DDL code representing the compatibility on the proper node and can add further goal flaws, disjunction flaws as well as preference flaws to the current search space node.

Unification is only applicable to tokens that have the same proposition. For the sake of compactness, we have introduced a new CSP constraint, that we call "multi-equals", defined as follows: given two sets of CSP variables $[x_0, \ldots, x_n]$ and $[y_0, \ldots, y_n]$, the multi-equals constraint is satisfied iff $[x_0 = y_0, \ldots, x_n = y_n]$.

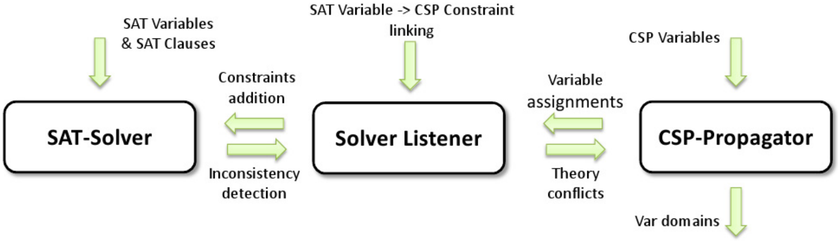


Fig. 2. Constraint reasoner architecture. The SAT solver controls most of the search aspects notifying the listener of variable assignments.

In order to identify timeline inconsistency flaws (aka *contention peaks*) inside the check consistency routine, tokens are sorted according to their earliest start time. In case of state variables, a set of Minimal Conflict Sets (MCSs) is identified by finding couples of overlapping tokens. In case of reusable resources, tokens are sequentially scrolled and collected in the current peak P until they overlap in time. Once a new token does not overlap with tokens in P, if total requirement of tokens in P exceeds resource capacity than P is added to the set of current peaks, all tokens that do not overlap with the last token are removed from P and the last token is added to P. The collecting process continues until all tokens are scrolled. For each peak, a set of Minimal Conflict Sets (MCSs) is identified by sorting tokens in decreasing order of their resource requirements and collecting MCSs in a lexicographic ordering. For each MCS, a timeline inconsistency flaw is added to the set of current flaws. Finally, timeline inconsistency flaws are solved simply creating a branch on the search tree for each of the possible orderings of the tokens constituting the MCS.

4.2 SAT-CSP Based Mixed Approach: J-TRE *(smt)*

To represent a partial plan and reason about it, in J-TRE*(smt)* we have pursued the idea of using a combination of SAT and CSP solving. As a starting point we have used an implementation of the well known MiniSAT solver [17] modified to endow it with capabilities for handling both preferences [18] and dynamic addition of variables and clauses. We will then let the SAT solving procedure to guide all the search process of both SAT and CSP problems in a Satisfiability Modulo Theory (SMT) fashion – e.g., [19]. Figure 2 shows the underlying structure of the constraint reasoner.

We can associate different CSP constraints to a SAT variable that "activates" them. Each time an activation variable is assigned true value by the SAT solver, the solver listener is notified and the correspondent constraints are dynamically added to the CSP performing propagation. The interplay works also the other way around: if a SAT variable goes from true to non assigned (the SAT solver is either backtracking or backjumping) then the corresponding constraints in the CSP are "deactivated" retracting them from the dynamic CSP that again is propagated to the previous situation. It is worth observing that the SAT solver manipulates variables according to a Last In First Out strategy facilitating efficiency of retraction in the correspondent dynamic CSP. A consequence of this is that we can safely reuse the J-TRE*(ac)* constraint reasoner to perform propagation of numeric variables. Furthermore, not all the SAT variables have a correspondent constraint. Those that are free from this connection can be used to model the causality in the planning problem or to add further constraints like those deriving from a disjunctive qualitative temporal network.

When the CSP propagation fails we have a *theory conflict*. In this case, the SAT solver is consistent but the correspondent theory represented by the CSP (the set of active constraints on CSP variables) is not. Similarly to the *lazy approach* in SMT, we add the information on the theory failure in the SAT representation by adding the negation of the conjunction of active constraints hence avoiding that the SAT solver reselect the same state later on. It is worth saying that the negation of a conjunction of literals can be transformed in a disjunction of negation by using De Morgan laws. In the SAT reasoner, this new clause is considered as a new "conflict clause" from which a no-good is generated and added the the representation after performing a backjumping step.

Giving preference for false values to each SAT variable allows us to minimize the number of active elements in the partial plan and, consequently, the number of active CSP constraints. Our extended CSP solver can now handle both disjunctive CSPs and domain causality through the SAT problem. If the SAT problem would become unsatisfiable then our extended CSP problem would have no solutions and the current search space node will become inconsistent.

Using the Hybrid Engine for Timeline Based Planning. So far, the planner collects flaws, selects one of them according to a selection strategy and solves it through the execution of DDL code which, in turns, adds new constraints among tokens and/or new tokens into the partial plan.

Having our own implementation of the SMT solver we can easily add further constraints in order to simplify the search procedure by pruning some search space. For example, we have modelled a disjunctive qualitative temporal network through SMT that would avoid qualitative inconsistencies. If we use $[\![x \leq y]\!]$ to denote the boolean variable which encodes the $x \leq y$ constraint, we can represent a disjunctive qualitative temporal network through the following constraints. For each triple of time-points t_0, t_1 and t_2, we add the following clauses representing the transitive closure: $(\neg [\![t_0 \leq t_1]\!] \vee \neg [\![t_1 \leq t_2]\!] \vee [\![t_0 \leq t_2]\!])$, $(\neg [\![t_1 \leq t_2]\!] \vee \neg [\![t_2 \leq t_0]\!] \vee [\![t_1 \leq t_0]\!])$ and $(\neg [\![t_0 \leq t_2]\!] \vee \neg [\![t_2 \leq t_1]\!] \vee [\![t_0 \leq t_1]\!])$.

***Flaw Management in J-TRE* (smt).** Each time a child is added to a node of the search space, the SMT of the parent node is copied into the child nodes together with all no-goods that have been learnt in previous search phases.

For what concerns goal flaws, we have two possible resolution strategies: *compatibility application* and *merging*. The flaw solver will generate a branch on the search space with two nodes representing the compatibility application on the first node and all possible merges on the second. Compatibility application is managed simply executing DDL code representing the compatibility on the proper node. Further state variable tokens and disjunctions can be added leading to new flaws on the current search space node.

Exploiting our multi-equals constraint we assign a SAT variable $u_0, \dots, u_i, \dots, u_n$, for each of the n tokens on the same timeline having the same proposition, to a multi-equals constraint that will ensure equality between goal token's arguments and target token's arguments. Whenever a variable u_i becomes true, the token is forced to unify with the correspondent token. The flaw solver will then add the resolution clause $(u_0 \lor \dots \lor u_n)$ to the SMT solver. By adding further clauses to the SMT solver we can link the merging of a token to the qualitative temporal network reasoning forcing the equality of the temporal points of the tokens. Assuming, for example, that the SAT variable u_0 represents the unification between tokens t_0 and t_1, we add the following clauses to the SMT: $(\neg u_0 \lor [\![t_0.s \leq t_1.s]\!])$, $(\neg u_0 \lor [\![t_1.s \leq t_0.s]\!])$, $(\neg u_0 \lor [\![t_0.e \leq t_1.e]\!])$ and $(\neg u_0 \lor [\![t_1.e \leq t_0.e]\!])$. Because unification does not lead to further compatibilities application, we always add preference for the unification case.

While the MCS extraction procedure is similar to the J-TRE*(ac)* case, we do not enqueue MCSs as common flaws because we can exploit SMT solver to directly solve them. We simply add a SAT clause asserting that the current state implies the disjunction of possible orderings of tokens. Let us consider, for example, the case in which we have an MCS composed by two tokens t_0 and t_1. The clause that will be added is $(\neg S \lor [\![t_0.e \leq t_1.s]\!] \lor [\![t_1.e \leq t_0.s]\!])$ where the symbol S is used to indicate the conjunction of all activation variables that in current state are active[2]. The clause $(\neg [\![t_0.e \leq t_1.s]\!] \lor \neg [\![t_1.e \leq t_0.s]\!])$ can also be added to improve performances.

4.3 Quantitative Time through Arc Consistency: J-TRE *(stp)*

Although really simple, J-TRE*(ac)* treats temporal points like standard numerical variables in a Single Source Shortest Path manner similar to [8]. The resulting temporal network doesn't allow almost any exploitation of information taken from constraint reasoning for search space pruning purposes resulting in an early search space explosion. With the intent to add more information in the constraint reasoning that can be used to prune search space, we have introduced J-TRE*(smt)* that maintains a disjunctive qualitative temporal network through a SAT encoding. We have seen that the J-TRE architecture is flexible enough to handle these two quite different technologies. A third attempt can be the use of a Simple Temporal Network [16] from which we can gain information that could possibly allow us to improve both flaw choosing and flaw resolution procedures.

In order to obtain an efficient All Pair Shortest Path behavior (namely the Simple Temporal Network we are looking for) without changing much from the J-TRE*(ac)* system, in J-TRE*(stp)* we have chosen to treat temporal points in a different way from standard numerical variables. For each couple of variables x_i and x_j we add an additional variable d_{ij} and a further constraint $d_{ij} = x_j - x_i$. The idea here is to use the d_{ij} to represent the distance between variables x_i and x_j. We could use this information to prune search space when solving scheduling flaws and/or merging in goal flows. Unfortunately, this addition is not enough for our simple AC-3 algorithm and distance

[2] Exploiting De Morgan laws, the symbol $\neg S$ is used to indicate the disjunction of negations of the literals of S. Being a disjunction of literals, with a little extension of terminology, we can use $\neg S$ inside clauses

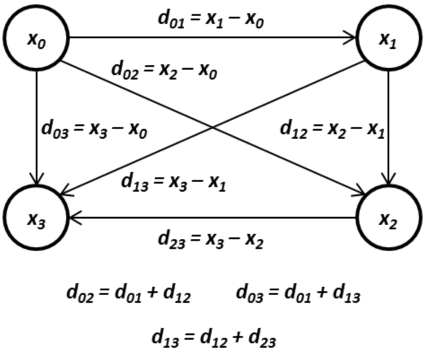


Fig. 3. An example of Simple Temporal Network with four time points in the J-TRE*(stp)* planner

information, often, is not propagated correctly. This problem can be easily solved by adding further constraints: for each couple of variables d_{ij} and d_{jk} we add the constraint $d_{ik} = d_{ij} + d_{jk}$. Figure 3 shows an example of Simple Temporal Network with four time points. Notice that since $d_{ij} = -d_{ji}$ we can economize on the number of variables and constraints.

It is worth underscoring how the use of numeric variables for maintaining distances between temporal points opens to interesting opportunities like linking the duration of an action to the amount of production (or consumption) of a given resource. By simply adding more variables and more constraints to our simple AC-3 based CSP solver we have obtained an efficient dynamic All Pair Shortest Path behavior providing constant time queries (once constraints have been propagated) about distance between temporal points.

***Flaw Management in J-TRE* (stp).** The flaw management in J-TRE*(stp)* is much more fruitful compared to J-TRE*(ac)* having the possibility to exploit information from the Simple Temporal Network. For example, it may happen that two overlapping tokens cannot be ordered in both ways or that some unifications cannot happen for temporal reasons. Moreover, if we find in a given search space node that a scheduling flaw cannot be solved in any way we can conclude that the entire node is an inconsistent one without caring of further flaws neither performing any constraint propagation.

It is worth underscoring that temporal constraints can no longer be defined among temporal variables but must be defined among distance variables. As an example, let us suppose we have two overlapping tokens t_i and t_j that result in a scheduling flaw. We know from querying distances between variables that $d_{t_i.e,t_j.s} \in [-inf, -1]$ and $d_{t_j.e,t_i.s} \in [-inf, +inf]$. From this information we deduce that we can schedule the two tokens only as $t_j \prec t_i$ (t_j before t_i). A simple constraint like $t_j.e \leq t_i.s$ wouldn't be enough as it wouldn't propagate temporal distance information. In order to schedule these tokens we need to force domain of variable $d_{t_j.e,t_i.s}$ to be greater or equal than 0. In a similar way, when unifying two tokens t_i and t_j, adding the multi-equals constraint is not enough as we also need to add constraint $d_{t_i.s,t_j.s} = 0$ as well as constraint $d_{t_i.e,t_j.e} = 0$.

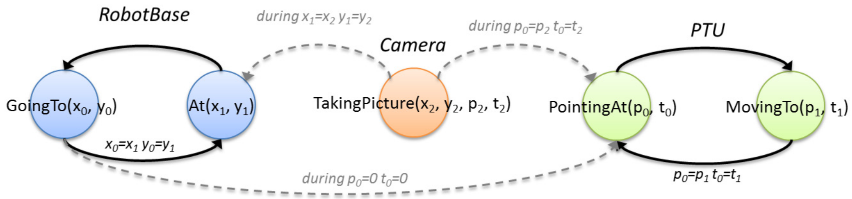


Fig. 4. State variables and resources describing the robotic platform and the communication channel availability

5 Evaluating the Planners

We now present a comparative evaluation of the three planners with respect to a robotic domain extracted from the GOAC project [20]. The domain is based on a robotic platform responsible for the movements, a payload represented by stereocameras mounted on a Pan-Tilt Unit, and a communication facility. We have devised two set of test domains: a simplified one (GOAC-*lite*) which focuses on the basic causality between platform and payload; a very realistic one (GOAC-*plus*) which represents a quite detailed picture of the scenario.

GOAC-*lite*. The basic domain contains the knowledge shown in Figure 4. The figure details the values that can be assumed by state variables and the legal value transitions in accordance with the mission requirements and the robot physics. To obtain a timeline-based specification of our robotic domain, we use three state variables, namely, *Robot-Base*, *PTU* and *Camera*. The robot can be in a position ($At(x, y)$) or moving towards a destination ($GoingTo(x, y)$). The PTU can assume a $PointingAt(pan, tilt)$ value if pointing a certain direction, while, when moving, it assumes a $MovingTo(pan, tilt)$ value. Finally, the camera can take a picture ($TackingPicture(x, y, p, t)$) of a given object in a position $\langle x, y\rangle$ with the PTU in $\langle pan, tilt\rangle$.

We can see how the causality concern the synchronization over time of (i) the PTU being in a safe position (pan and tilt at 0) while the robot is moving, (ii) to take a picture the robot must be still in one of the requested locations and the PTU should be pointing at the related direction. Compatibilities express the following: (i) $GoingTo(x, y)$ must occur during $PointingAt(0, 0)$; (ii) *TakingPicture(x, y, pan, tilt)* must occur during $At(x, y)$ and $PointingAt(pan, tilt)$;

The three planners were run on problems having an increasing number of temporally unconstrained “TakingPicture” goals (a bunch of targets to return picture of). Figure 5 shows execution time (in milliseconds) of our benchmark problem (the number of $TakingPicture$ goals are on the abscissas). We can see that the overhead added by the SAT encoding of the qualitative temporal network (the J-TRE*(smt)* planner) exceeds the benefits of search space pruning that derives from it. The overhead added in the J-TRE*(stp)* planner, on the other side, allows good search space pruning and outperforms other planners even on small problem instances.

GOAC-*plus*. The second domain is a rather detailed version of the real domain. We have modeled: the visibility windows to communicate with the relay satellite that sends

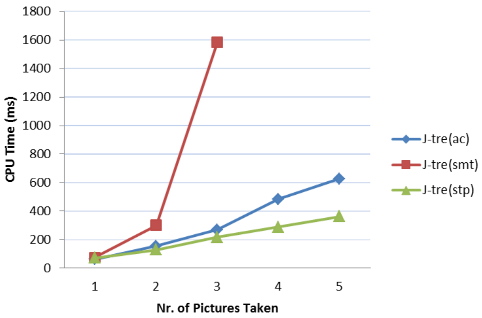


Fig. 5. Comparative results on the GOAC-*lite* domain

data to Earth, two on board resources, namely, memory and power usage. Notice that while GOAC-*lite* was a purely planning problem, because the key aspect was the causal combination of constraints among platform-camera-PTU, the GOAC-*plus* is a quite challenging planning and scheduling problem.

During operation, the rover should follow some rules to maintain safe and effective configurations. Namely, the following conditions must hold during the overall mission: (i) while the robot is moving, the PTU must be in the safe position (pan and tilt at 0) and 40W of power are required; (ii) the robotic platform can take a picture only if the robot is still in one of the requested locations while the PTU is pointing to the related direction and if an adequate amount of on board memory is available to store the picture; (iii) once a picture has been taken, the rover has to communicate the picture to the base station; (iv) while communicating the rover has to stay still, 60W of power are requested and the memory is released of the amount of transmitted file; (v) while communicating, the orbiter needs to be visible.

In Figure 6 we detail the values that can be assumed by these state variables and the legal value transitions in accordance with the mission requirements and the robot physics. The robot can be in a position $(At(x,y))$ or moving towards a destination $(GoingTo(x,y))$. The PTU can assume a $PointingAt(pan,tilt)$ value if pointing a certain direction, while, when moving, it assumes a $MovingTo(pan,tilt)$ value. The camera either takes a picture of a given object in a position $\langle x,y\rangle$ with the PTU in $\langle pan,tilt\rangle$ $(TakingPicture(file{-}id,x,y,pan,tilt))$ or is idle. Similarly, the communication facility can be either operative and dumping a given file $(Communicating(file-id))$ or idle. The reusable resource *Power* represents consumed power in time while the consumable resource *Memory* represents memory consumption in time. Additionally, one external resource, the *HRDL*, represents contingent events, i.e., the communication channel availability.

As an example, a possible mission action sequence can be the following: navigate to one of the requested locations, move the PTU pointing at the requested direction, take a picture, then, communicate the image to the orbiter during the next available visibility window, put back the PTU in the safe position and, finally, move to the following requested location. Once all the locations have been visited and all the pictures have been communicated, the mission is considered successfully completed.

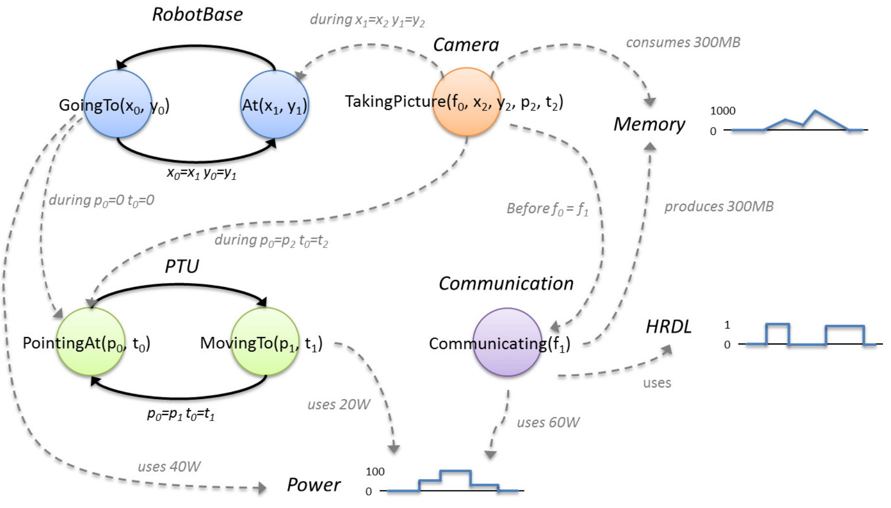


Fig. 6. State variables and resources describing the robotic platform and the communication channel availability

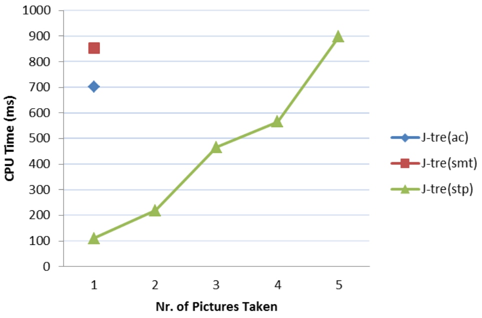


Fig. 7. Execution time (in milliseconds) for the GOAC-*plus* domain (for increasing number of $TakingPicture$ goals)

Given an initial situation in which the robot is at location $\langle 0, 0 \rangle$ and the pan-tilt is oriented to $\langle 0, 0 \rangle$, we scale the problem by adding different $TakingPicture$ goals. Figure 7 shows execution time (in milliseconds) of our benchmark problem (the number of $TakingPicture$ goals on the abscissas) for all the presented planners. We can observe how the J-TRE*(stp)* planner scales quite well with respect to all the problem instances. On the contrary, the intrinsic complexity of the problem overwhelms both the J-TRE*(ac)* and the J-TRE*(smt)* planners that only solve the single smaller instance of the problem.

6 Conclusions

Temporal flexibility required in controlling mechanisms in real-time (i.e., robotics), interacting with agents requirements as well as uncertainty of real world domains, are just some of the arguments that are leading to the progressive exploration of different planning methodologies and to the extensions of most classic ones.

Timeline based planning constitutes an intuitive alternative to classical planning approaches by identifying relevant domain components evolving in time. Although attractive from a temporal flexibility point of view, these kind of planners have to cope with performance issues due to the complexity that derives from their expressiveness.

In this paper we have proposed an architecture that filters out common elements from timeline-based planners with the intent to focus the attention to underlying constraint reasoners that could lead to different possible flaw selection and flaw resolution strategies. In so doing we have discovered the superiority of a particular approach over the other proposed ones in the context of the given domain. However we are confident that other approaches can be improved and can possibly outperform the actual results in different domains.

Some points are still open. The heuristics for flaw selection and flaw resolution strategies are still relatively poor to compete in performance with state-of-the-art classical planners. The thrust to classical planning given by GRAPHPLAN and the consequent development of heuristic based search is something that is still missing in the timeline-based area.

Acknowledgements. Authors have been partially supported by EU under the PANDORA project (FP7.225387) and by MIUR under the PRIN project 20089M932N.

References

1. Jonsson, A., Morris, P., Muscettola, N., Rajan, K., Smith, B.: Planning in Interplanetary Space: Theory and Practice. In: Proceedings of the Fifth Int. Conf. on AI Planning and Scheduling, AIPS 2000 (2000)
2. Chien, S., Tran, D., Rabideau, G., Schaffer, S., Mandl, D., Frye, S.: Timeline-Based Space Operations Scheduling with External Constraints. In: Proc. of the 20th Int. Conf. on Automated Planning and Scheduling, ICAPS 2010 (2010)
3. Ghallab, M., Laruelle, H.: Representation and Control in IxTeT, a Temporal Planner. In: Proceedings of the 2nd Int. Conf. on AI Planning and Scheduling, AIPS 1994, pp. 61–67 (1994)
4. Fratini, S., Pecora, F., Cesta, A.: Unifying Planning and Scheduling as Timelines in a Component-Based Perspective. Archives of Control Sciences 18(2), 231–271 (2008)
5. Cesta, A., Cortellessa, G., Fratini, S., Oddi, A.: Developing an End-to-End Planning Application from a Timeline Representation Framework. In: Proc. of the 21st Conf. on Innovative Applications of Artificial Intelligence, IAAI 2009 (2009)
6. Frank, J., Jonsson, A.: Constraint-Based Attribute and Interval Planning. Constraints 8(4), 339–364 (2003)
7. Fox, M., Long, D.: PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains. Journal of Artificial Intelligence Research 20, 61–124 (2003)
8. Cesta, A., Oddi, A.: Gaining Efficiency and Flexibility in the Simple Temporal Problem. In: TIME 1996: Proc. of the 3rd Int. Workshop on Temporal Representation and Reasoning (1996)
9. Laborie, P.: Algorithms for propagating resource constraints in ai planning and scheduling: existing approaches and new results. Artificial Intelligence 143, 151–188 (2003)

10. Weld, D.S.: An Introduction to Least Commitment Planning. AI Magazine 15(4), 27–61 (1994)
11. Blum, A., Furst, M.L.: Fast Planning Through Planning Graph Analysis. Artificial Intelligence 90, 281–300 (1997)
12. Bonet, B., Geffner, H.: Planning as heuristic search. Artificial Intelligence 129, 5–33 (2001)
13. Muscettola, N.: HSTS: Integrating Planning and Scheduling. In: Zweben, M., Fox, M.S. (eds.) Intelligent Scheduling. Morgan Kaufmann (1994)
14. Allen, J.F.: Maintaining Knowledge about Temporal Intervals. Commun. ACM 26(11), 832–843 (1983)
15. Bernardini, S., Smith, D.: Developing domain-independent search control for EUROPA2. In: Proceedings of the Workshop on Heuristics for Domain-independent Planning at ICAPS 2007 (2007)
16. Dechter, R.: Constraint Processing. Morgan Kaufmann (2003)
17. Eén, N., Sörensson, N.: An Extensible SAT-solver. In: Giunchiglia, E., Tacchella, A. (eds.) SAT 2003. LNCS, vol. 2919, pp. 502–518. Springer, Heidelberg (2004)
18. Di Rosa, E., Giunchiglia, E., Maratea, M.: Solving Satisfiability Problems with Preferences. Constraints 15(4), 485–515 (2010)
19. Sebastiani, R.: Lazy Satisability Modulo Theories. JSAT 3(3-4), 141–224 (2007)
20. Ceballos, A., Bensalem, S., Cesta, A., de Silva, L., Fratini, S., Ingrand, F., Ocon, J., Orlandini, A., Py, F., Rajan, K., Rasconi, R., van Winnendael, M.: A Goal-oriented Autonomous Controller for Space Exploration. In: Proc. 11th Symposium on Advanced Space Technologies in Robotics and Automation, ASTRA 2011 (2011)

Efficient Spatial Reasoning with Rectangular Cardinal Relations and Metric Constraints

Angelo Montanari[1], Isabel Navarrete[2], Guido Sciavicco[2], and Alberto Tonon[3]

[1] Department of Mathematics and Computer Science. University of Udine, Italy
[2] Department of Information Engineering. University of Murcia, Spain
[3] eXascale Infolab. University of Fribourg, Switzerland
angelo.montanari@uniud.it, {inava,guido}@um.es,
alberto.tonon@unifr.ch

Abstract. In many real-world applications of knowledge representation and reasoning formalisms, one needs to cope with a number of spatial aspects in an integrated and efficient way. In this paper, we focus our attention on the so-called Rectangular Cardinal Direction calculus for qualitative spatial reasoning on cardinal relations between rectangles whose sides are parallel to the axes of a fixed reference system. We show how to extend its convex tractable fragment with metric constraints preserving tractability. The resulting formalism makes it possible to efficiently reason about spatial knowledge specified by one qualitative constraint network and two metric networks (one for each spatial dimension). In particular, it allows one to represent definite or imprecise knowledge on directional relations between rectangles and to derive additional information about them, as well as to deal with metric constraints on the height/width of a rectangle or on the vertical/horizontal distance between the sides of two rectangles. We believe that the formalism features a good combination of simplicity, efficiency, and expressive power, making it adequate for spatial applications like, for instance, web-document query processing and automatic layout generation.

Keywords: Qualitative spatial reasoning, Quantitative spatial reasoning, Cardinal direction relations, Constraint satisfaction problems.

1 Introduction

Qualitative spatial representation and reasoning play an important role in various areas of computer science such as, for instance, geographic information systems, spatial databases, document analysis, layout design, and image retrieval. Different aspects of space, such as direction, topology, size, and distance, which must be dealt with in a coherent way in many real-world applications, have been modeled by different formal systems (see [5] for a survey). For practical reasons, a bidimensional space is commonly assumed, and spatial entities are represented by points, boxes, or polygons with a variety of shapes, depending on the required level of detail.

Information about spatial configurations is usually specified by constraint networks describing the allowed binary relations between pairs of spatial variables. The main problem in qualitative spatial reasoning is to decide whether or not a given network

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 234–249, 2013.

has a solution, that is, to establish whether or not there exists an assignment of domain values to variables that satisfies all constraints (*consistency checking*).

Cardinal relations are directional relations that allow one to specify how spatial objects are placed relative to each other either by making use of a fixed reference system, e.g., to say that an object is to the "north" or "southwest" of another one in a geographic space, or, alternatively, by exploiting directions as "above" or "below and left" in a local space. The most expressive formalism with cardinal relations between plane regions is the *Cardinal Direction calculus* (*CD-calculus* for short) [10,14,22]. The consistency problem for the CD-calculus is NP-complete; moreover, in [12], it has been shown that there exists no tractable fragment of it containing all basic constraints. Such a restriction is a serious limitation when we have to deal with incomplete or indefinite information in spatial applications.

A restricted version of the CD-calculus, called *Rectangular Cardinal Direction calculus* (*RCD-calculus*), has been introduced in [19,18], where cardinal relations are defined only between rectangles whose sides are parallel to the axes of the Euclidean plane. Rectangles of this type (aka *boxes*) can be seen as *minimum bounding rectangles (MBRs)* that enclose plane regions (the actual spatial objects). On the one hand, approximating regions by rectangles implies a loss of accuracy in the representation of the relative direction between regions; on the other hand, reasoning tasks become more efficient. The RCD-calculus has a strong connection with the *Rectangle Algebra* (RA) [2], which can be viewed as a bidimensional extension of *Interval Algebra* (IA), the well-known temporal formalism for dealing with qualitative binary relations between time intervals [1]. A tractable fragment of the RCD-calculus, called *convex RCD-calculus*, has been identified in [18]. It includes all basic relations and a large number of disjunctive relations, making it possible to represent and reason about indefinite information efficiently.

In this paper, we extend the convex RCD-calculus with metric constraints. Metric constraints between points over a dense linear order have been dealt with by the Temporal Constraint Satisfaction Problem formalism (TCSP) [7]. In such a formalism, one can restrict the admissible values for the distance between a pair of points to a finite set of ranges. If each constraint consists of one range only, we get a tractable fragment of TCSP, called Simple Temporal Problem formalism (STP). In the following, we propose a metric extension to the convex RCD-calculus that allows one to represent available knowledge on directional relations between rectangles and to derive additional information about them, as well as to deal with metric constraints on the height/width of a rectangle or on the vertical/horizontal distance between rectangles. We will show that the resulting formalism is expressive enough to capture various scenarios of practical interest and still computationally affordable.

The rest of the paper is organized as follows. In Section 2, we provide background knowledge on qualitative calculi and we shortly recall Interval Algebra and Rectangle Algebra. In Section 3, we introduce the RCD-calculus and its convex fragment. In Section 4, we extend the convex RCD-calculus with metric features, and we devise a sound and complete polynomial algorithm for consistency checking. In Section 5, we apply the proposed formalism to a case study in the domain of web-document layout design. Conclusions provide an assessment of the work and outline future research directions.

2 Background

In this section, we briefly review basic notions on constraint networks and the main calculi regarding qualitative relations on points, intervals and rectangles.

Temporal (resp., spatial) knowledge is commonly represented in a qualitative calculus by means of a *qualitative network* consisting of a complete constraint-labeled digraph $N = (V, C)$, where $V = \{v_1, \ldots, v_n\}$ is a finite set of variables, interpreted over an infinite domain D, and the labeled edges in C specify the constraints defining qualitative temporal (resp., spatial) configurations. An edge from v_i to v_j labeled with R corresponds to the *constraint* $v_i \, R \, v_j$, where R denotes a binary relation over D which restricts the possible values for the pair of variables (v_i, v_j). The full set of relations of the calculus is usually taken as the powerset $2^{\mathcal{B}}$, where $\mathcal{B}$ is a finite set of binary *basic relations* that forms a partition of $D \times D$. Thus, a relation $R \in 2^{\mathcal{B}}$ is of the form $R = \{r_1, \ldots, r_m\}$, where each r_i is a basic relation, and R represents the union of the basic relations it contains. If $m = 1$, we call R a *basic relation*; otherwise ($m > 1$), we call it a *disjunctive relation*. A special case of disjunctive relation is the *universal relation*, denoted by '?', which contains all the basic relations. A *basic constraint* $v_i\{r\}v_j$ expresses definite knowledge about the values that the two variables v_i, v_j can take, while a *disjunctive constraint* $v_i\{r_1, \ldots, r_m\}v_j$ expresses indefinite or imprecise knowledge about these values. In particular, the *universal constraint* $v_i \, ? \, v_j$ states that the relation between v_i an v_j is totally unknown. From a logical point of view, a disjunctive constraint $v_i\{r_1, \ldots, r_m\}v_j$ can be viewed as the disjunction $v_i \, \{r_1\} \, v_j \vee \ldots \vee v_i \, \{r_m\} \, v_j$.

An *instantiation* of the constraints of a qualitative network N is a mapping ι representing an assignment of domain values to the variables of N. A constraint $v_i \, R \, v_j$ is said to be *satisfied* by an instantiation ι if the pair $(\iota(v_i), \iota(v_j))$ belongs to the binary relation represented by R. A *consistent instantiation*, or *solution*, of a network is an assignment of domain values to variables satisfying all the constraints. If such a solution exists, then the network is consistent, otherwise it is inconsistent.

The main reasoning task in qualitative reasoning is *consistency checking*, which amounts to deciding if a network is consistent. If all relations are considered, consistency checking is usually NP-hard. Hence, finding subsets of $2^{\mathcal{B}}$ for which consistency checking turns out to be polynomial (*tractable subsets*) is an important issue to address. Another common task in qualitative reasoning is computing the unique *minimal network* equivalent to a given one by determining, for each pair of variables, the *strongest* relation *(minimal relation*) entailed by the constraints of the network. It can be easily shown that each basic relation in a minimal network is *feasible*, i.e., it participates in some solution of the network. To deal with these tasks, constraint propagation techniques are usually exploited [25,23]. The most prominent method for constraint propagation in qualitative temporal reasoning is the so-called *path-consistency algorithm*, PC-algorithm for short [15]. Such an algorithm refines relations by successively applying the operation $R_{ij} \leftarrow R_{ij} \cap (R_{ik} \circ R_{kj})$ for every triple of variables (v_i, v_k, v_j), until a stable network is reached, where R_{ij}, R_{ik}, R_{kj} are the relations constraining the pair of variables (v_i, v_j), (v_i, v_k), (v_k, v_j), respectively ($\circ$ stands for the composition of relations). If the empty relation is obtained during the process, then the input network is inconsistent; otherwise, we can conclude that the output network is *path consistent*,

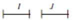
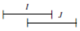
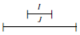
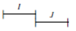
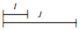
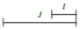
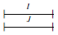
Relation	Symbol	Inverse	Illustration	Meaning
I before J	b	bi		$I^- < I^+ < J^- < J^+$
I overlaps J	o	oi		$I^- < J^- < I^+ < J^+$
I during J	d	di		$J^- < I^- < I^+ < J^+$
I meets J	m	mi		$I^- < I^+ = J^- < J^+$
I starts J	s	si		$I^- = J^- < I^+ < J^+$
I finishes J	f	fi		$J^- < I^- < I^+ = J^+$
I equals J	e	e		$I^- = J^- < I^+ = J^+$

Fig. 1. Basic relations of the Interval Algebra

which does not necessarily imply that it is consistent. In some special cases, the PC-algorithm can be used to decide the consistency of a qualitative network and to get the minimal one.

2.1 Interval Algebra and Point Algebra

Allen's *Interval Algebra* (IA) allows one to constrain the relative position of two time intervals [1]. An interval I is usually interpreted as a closed interval over the rational numbers $[I^-, I^+]$, whose *endpoints* I^- and I^+ satisfy the relation $I^- < I^+$. Let $\mathcal{B}_{ia}$ be the set of the thirteen *basic interval relations* capturing all possible ways to order the four endpoints of two intervals, usually denoted by the symbols $b, o, d, m, s, f, e, bi, oi,$ di, mi, si, and fi. The semantics of basic IA-relations is defined in terms of ordering relations between the endpoints of the intervals, as shown in Figure 1. Notice that, given a basic relation r between two intervals I and J, the inverse relation ri is defined by simply exchanging the roles of I and J (see Figure 1). IA can be viewed as a constraint algebra defined by the power set $2^{\mathcal{B}_{ia}}$ and the operations of intersection, inverse $(^{-1})$, and composition $(\circ)$ of relations.

IA subsumes *Point Algebra* (PA) [25], a simpler qualitative calculus whose binary relations specify the relative position of pairs of time points. PA binary relations are $<, >, =$ (basic) and $\leq, \geq, \neq, ?$ (disjunctive), plus the empty relation. The endpoint relations defining a basic IA-relation (Figure 1) are basic relations of PA.

2.2 Rectangle Algebra

Rectangle Algebra (RA), proposed by Balbiani et al. (1998), is an extension of IA to a bidimensional space. We assume here the domain of RA to consist of the set of rational rectangles whose sides are parallel to the axes of the Euclidean plane. To avoid a notational overload, with a little abuse of notation, hereafter we will denote by a, b both rectangles in the domain of RA and constraint (rectangle) variables. A rectangle a is completely characterized by a pair of intervals (a_x, a_y), where a_x and a_y are the projections of a onto the x- and y-axis, respectively. We call $\mathcal{B}_{ra}$ the set of basic relations of RA, which is obtained by considering all possible pairs of basic IA-relations.

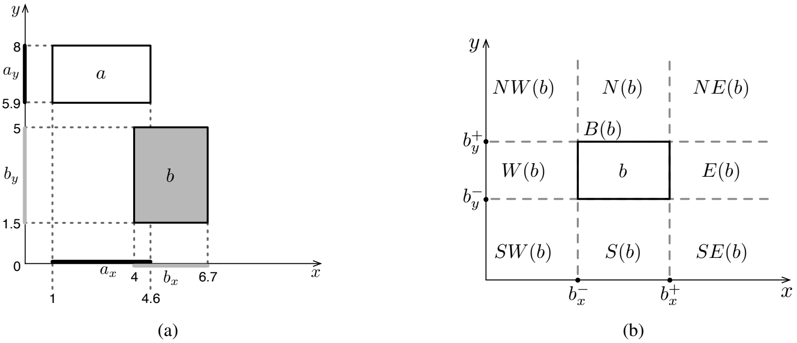


Fig. 2. (a) An illustration of the RA-relation $a\{(o, bi)\}b$. The corresponding RCD-relation is $a\,\{NW{:}N\}\,b$. (b) Cardinal tiles with respect to rectangle b.

Hence, a basic RA-relation r is denoted by a pair $r = (t, t')$ of basic IA-relations, representing the set of pairs of *rectangles* (a, b) such that $a\,(t, t')\,b$ holds if and only if, by definition, $a_x\,t\,b_x$ and $a_y\,t'\,b_y$ hold. Given a basic RA-relation $r = (t, t')$, let $t = \pi_x(r)$ and $t' = \pi_y(r)$ be the x- and y-projection of r, respectively.

Example 1. Figure 2-(a) shows a spatial realization of $a\,\{(o, bi)\}\,b$. Note that $\pi_x(o, bi) = o$, $\pi_y(o, bi) = bi$, a_x overlaps b_x, and a_y is after b_y. The left endpoints of the intervals assigned to a_x and a_y (1 and 5.9, respectively) and their right endpoints (4.6 and 8, respectively) are the coordinates of the lower-left and upper-right vertices of the given instantiation of a, respectively. The same for b. Thus, the values assigned to the endpoints of the projections of a and b represent an assignment for a and b that satisfies the constraint $a\,\{(o, bi)\}\,b$.

In the case of an arbitrary RA-relation $R \in 2^{\mathcal{B}_{ra}}$, the *projections of* R are defined as follows:

$$\pi_x(R) = \{\pi_x(r) \mid r \in R\} \quad \pi_y(R) = \{\pi_y(r) \mid r \in R\}.$$

Notice that, in general, $\pi_x(R) \times \pi_y(R)$ may be different from R or, equivalently, we may have $\pi_x(R_1) = \pi_x(R_2)$ and $\pi_y(R_1) = \pi_y(R_2)$ for some $R_1 \neq R_2$.

The mappings π_x and π_y can be generalized to RA-networks. We define the projections π_x and π_y of an RA-network $N = (V, C)$ as the two IA-networks $\pi_x(N) = (V_x, C_x)$ and $\pi_y(N) = (V_y, C_y)$, where V_x, V_y are the sets of interval variables corresponding to the rectangle variables in V and the set of IA-constraints C_x (resp., C_y) is obtained by replacing each relation R_{ij} in C by $\pi_x(R_{ij})$ (resp., by $\pi_y(R_{ij})$).

2.3 Convex Subalgebras

The consistency problem for both IA and RA is known to be NP-complete. Several tractable fragments of both calculi have been identified in the literature. In this paper, we focus our attention on convex tractable subsets of IA [24] and RA [2], which consist of the set of *convex IA-relations* and *convex RA-relations*, respectively. Convex relations are those relations that can be equivalently expressed as a set of convex PA-constraints (all PA-relations except $\neq$ are allowed) between the endpoints of interval

Table 1. Translation of convex PA-constraints to STP-constraints via the $toSTP$ mapping

Convex PA relation	STP constraint
$p_i < p_j$	$p_j - p_i \in]0, +\infty[$
$p_i \leq p_j$	$p_j - p_i \in [0, +\infty[$
$p_i = p_j$	$p_j - p_i \in [0, 0]$
$p_i > p_j$	$p_j - p_i \in]-\infty, 0[$
$p_i \geq p_j$	$p_j - p_i \in]-\infty, 0]$
$p_i \, ? \, p_j$	$p_j - p_i \in]-\infty, +\infty[$

variables (convex IA-relations) or between the endpoints of the projections of rectangle variables (convex RA-relations). It is worth to mention that a convex RA-relation is equivalently characterized as an RA-relation which can be obtained as the Cartesian product of two convex IA-relations. Both the consistency and the minimality problems in the convex fragments of PA, IA, and RA can be solved in $O(n^3)$, where n is the number of variables of the input network.

2.4 Simple Temporal Problem

The Simple Temporal Problem (STP) formalism was introduced in [7] to process metric information about time points. Formally, an STP is specified by a constraint network $S = (P, M)$, where P is a set of n point variables, whose values range over a dense domain (which we assume to be $\mathbb{Q}$), and M is a set of *binary metric* constraints over P. A metric constraint $M_{ij} = [q, q']$ (open and semi-open intervals can be used as well), with $q, q' \in \mathbb{Q}$, on the distance between (the values of) $p_i, p_j \in P$ states that $p_j - p_i \in [q, q']$, or, equivalently, that $q \leq p_j - p_i \leq q'$. Hence, the constraint M_{ij} defines the set of possible values for the distance $p_j - p_i$. In the constraint graph associated to S, $M_{ij} = [q, q']$ is represented by an edge from p_i to p_j labeled by the rational interval $[q, q']$. *Unary metric* constraints restricting the domain of a point variable p_i can be encoded as binary constraints between p_i and a special starting-point variable with a fixed value, e.g., 0. The *universal constraint* is $] - \infty, +\infty[$. The operations of composition ($\circ$) and inverse ($^{-1}$) of metric constraints are computed by means of interval arithmetic, that is, $[q_1, q_2] \circ [q_3, q_4] = [q_1 + q_3, q_2 + q_4]$ and $[q_1, q_2]^{-1} = [-q_2, -q_1]$. Intersection of constraints (intervals) is defined as usual.

Assuming such an interpretation of the operations of composition, inverse, and intersection, in [7] Dechter et al. show that any path-consistency algorithm can be exploited to compute the minimal STP equivalent to a given one, if any (if an inconsistency is detected, the algorithm returns an empty network). In the following, we will denote such an algorithm by $\mathbf{PC_{stp}}$. Making use of such a result, in [16], Meiri proposes a formalism to combine qualitative constraints between points and intervals with (possibly disjunctive) metric constraints between points (as in TCSP). An easy special case arises when only convex PA-constraints and STP-constraints are considered. Convex PA-constraints can be encoded as STP-constraints by means of the $toSTP$ translation function described in Table 1. The following result can be found in [16].

Theorem 1. *Let N be a network with convex PA-constraints and STP-constraints. If N is path-consistent, then N is also consistent and its metric constraints are minimal.*

$\mathbf{PC_{stp}}$ can thus be used to decide the consistency of a network N satisfying the conditions of the above theorem. To this end, it suffices to encode PA-constraints into equivalent STP-constraints.

3 Rectangular Cardinal Direction Calculus

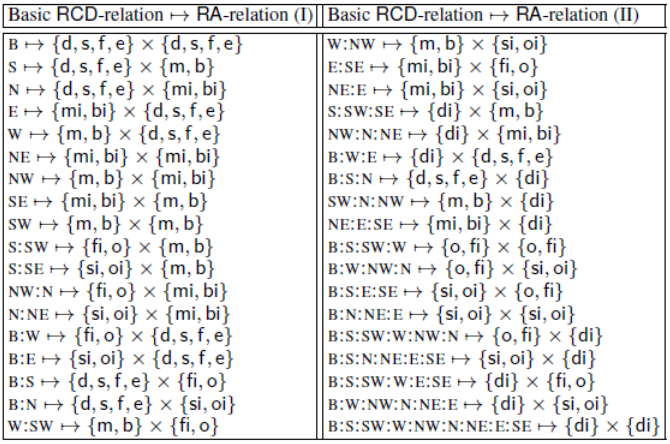
The Rectangular Cardinal Direction calculus (*RCD-calculus*) [19,18] deals with cardinal direction relations between rectangles. It can be viewed as a restricted version of the CD-calculus over the domain of regular regions [10,14,22], which includes all rectangles aligned to the axes. The domain of the CD-calculus is the same as that of RA. Let b be a *reference* rectangle. We denote by b_x^- and b_x^+ (resp., b_y^- and b_y^+) the left and the right endpoint of the projection of b onto the x-axis (resp., y-axis), respectively. The straight lines $x = b_x^-$, $x = b_x^+$, $y = b_y^-$, $y = b_y^+$ divide the plane into nine tiles $\tau_i(b)$, with $1 \leq i \leq 9$, as shown in Figure 2-(b), where τ_i is a *tile symbol* from the set $TS = \{B, S, SW, W, NW, N, NE, E, SE\}$, denoting the cardinal directions in the Bounds of, to the South of, to the SouthWest of, to the West of, to the NorthWest of, to the North of, to the NorthEast of, to the East of, and to the SouthEast of, respectively.

Definition 1. *A* basic rectangular cardinal relation *(basic RCD-relation) is denoted by a* tile string $\tau_1{:}\tau_2{:}\ldots{:}\tau_k$, *where $\tau_i \in TS$, for $1 \leq i \leq k$, such that $a\ \tau_1{:}\tau_2{:}\ldots{:}\tau_k\ b$ holds iff for all $\tau_i \in \{\tau_1, \tau_2, \ldots, \tau_k\}$, $a^\circ \cap \tau_i(b) \neq \varnothing$, and for all $\tau_i \in TS \setminus \{\tau_1, \tau_2, \ldots, \tau_k\}$, $a^\circ \cap \tau_i(b) = \varnothing$, where a° is the interior of a. A* rectangular cardinal relation *(RCD-relation) is represented by a set $R = \{r_1, \ldots, r_m\}$, where each r_i is a basic RCD-relation.*

The set $\mathcal{B}_{rcd}$ of basic RCD-relations consists of 36 elements (see Figure 3). Qualitative networks with labels in $2^{\mathcal{B}_{rcd}}$, as well as the consistency problem for such networks, are defined in the standard way.

3.1 RCD-Calculus and Rectangle Algebra

The relationships between the RCD-calculus and the Rectangle Algebra have been systematically investigated by Navarrete et al. in [18]. Consider, for instance, the RCD-constraint $a\,\{NW{:}N\}\,b$. A possible instantiation of such a constraint is depicted in Figure 2-(a). The same pair of rectangles can be viewed as an instance of the RA-constraint $a\,\{(o, bi)\}\,b$ as well. However, there exists another possible instantiation of the constraint $a\,\{NW{:}N\}\,b$ that satisfies the RA-constraint $a\,\{(o, mi)\}\,b$. In general, for a given RCD-relation there exist more than one corresponding RA-relations, while for a given RA-relation there exists exactly one corresponding RCD-relation. This is due to the fact that RCD-relations are coarser than RA-relations. As an example, the RCD-calculus does not allow one to precisely state that two given rectangles are externally connected or strictly disconnected, or to constrain their sides to be (or to be not) vertically (resp., horizontally) aligned. As a general rule, given an RCD-relation, we



Basic RCD-relation ↦ RA-relation (I)	Basic RCD-relation ↦ RA-relation (II)
B ↦ {d, s, f, e} × {d, s, f, e}	W:NW ↦ {m, b} × {si, oi}
S ↦ {d, s, f, e} × {m, b}	E:SE ↦ {mi, bi} × {fi, o}
N ↦ {d, s, f, e} × {mi, bi}	NE:E ↦ {mi, bi} × {si, oi}
E ↦ {mi, bi} × {d, s, f, e}	S:SW:SE ↦ {di} × {m, b}
W ↦ {m, b} × {d, s, f, e}	NW:N:NE ↦ {di} × {mi, bi}
NE ↦ {mi, bi} × {mi, bi}	B:W:E ↦ {di} × {d, s, f, e}
NW ↦ {m, b} × {mi, bi}	B:S:N ↦ {d, s, f, e} × {di}
SE ↦ {mi, bi} × {m, b}	SW:N:NW ↦ {m, b} × {di}
SW ↦ {m, b} × {m, b}	NE:E:SE ↦ {mi, bi} × {di}
S:SW ↦ {fi, o} × {m, b}	B:S:SW:W ↦ {o, fi} × {o, fi}
S:SE ↦ {si, oi} × {m, b}	B:W:NW:N ↦ {o, fi} × {si, oi}
NW:N ↦ {fi, o} × {mi, bi}	B:S:E:SE ↦ {si, oi} × {o, fi}
N:NE ↦ {si, oi} × {mi, bi}	B:N:NE:E ↦ {si, oi} × {si, oi}
B:W ↦ {fi, o} × {d, s, f, e}	B:S:SW:W:NW:N ↦ {o, fi} × {di}
B:E ↦ {si, oi} × {d, s, f, e}	B:S:N:NE:E:SE ↦ {si, oi} × {di}
B:S ↦ {d, s, f, e} × {fi, o}	B:S:SW:W:E:SE ↦ {di} × {fi, o}
B:N ↦ {d, s, f, e} × {si, oi}	B:W:NW:N:NE:E ↦ {di} × {si, oi}
W:SW ↦ {m, b} × {fi, o}	B:S:SW:W:NW:N:NE:E:SE ↦ {di} × {di}

Fig. 3. Translation from basic RCD-relations to RA-relations via the $toRA$ mapping

can always determine the strongest RA-relation it implies. As an example, the strongest RA-relation implied by $NW{:}N$ is $\{fi, o\} \times \{mi, bi\}$. Notice that such an RA-relation, which is entailed by a basic RCD-relation, is not a basic RA-relation.

The weaker expressive power of the RCD-calculus with respect to RA is not necessarily a problem. As an example, if we are interested in pure cardinal information only, the expressiveness of RCD-relations suffices. Moreover, the constraint language of the RCD-calculus is closer to the natural language than the one of the RA. For example, stating that "rectangle a lies partly to the northwest and partly to the north of b" ($a\,\{NW{:}N\}\,b$) is much more natural than stating that "the x-projection of a is overlapping or finished by the x-projection of b, and the y-projection of b is $\ldots$" ($a\{fi, o\} \times \{mi, bi\}b$).

Figure 3 describes a translation function, called $toRA$, to map a basic RCD-relation into the strongest entailed RA-relation. This mapping can be extended to translate arbitrary relations, constraints, and networks of the RCD-calculus to their RA counterparts, preserving consistency. More precisely, given a disjunctive relation R, $toRA(R)$ is obtained as the union of the translation of the basic relations in R, while, given an RCD-network $N = (V, C)$, the corresponding RA-network $toRA(N)$ is obtained by replacing each relation R_{ij} in C by $toRA(R_{ij})$. As the following theorem states, to decide the consistency of an RCD-network N, one can compute $toRA(N)$ and then apply to it any algorithm for deciding the consistency of RA-networks [18].

Theorem 2. *An RCD-network N is consistent if and only if $toRA(N)$ is consistent.*

3.2 The Convex Fragment of the RCD-Calculus

In [18], the authors show that the consistency problem for the RCD-calculus is NP-complete and they identify a large tractable subset of RCD-relations. Such a fragment, called *convex RCD-calculus*, consists of all and only the RCD-relations R whose translation $toRA(R)$ is a convex RA-relation. It is possible to show that there exist 400

convex RCD-relations. As we already pointed out, the convex subclasses of IA, PA, and RA are tractable. By exploiting the connection between these subclasses and the convex RCD-calculus, an $O(n^2)$ algorithm for consistency checking of convex RCD-networks has been proposed in [18]. In particular, it benefits from the following result about RA, stated in [2].

Theorem 3. *Let N be a convex RA-network. N is consistent if and only if its projections $\pi_x(N)$ and $\pi_y(N)$ are consistent.*

4 Convex-Metric RCD-Calculus

In this section, we propose a tractable metric extension of the convex RCD-calculus, called *convex-metric RCD-calculus* (*cmRCD-calculus*) to represent and to reason about both qualitative cardinal constraints between rectangles and metric constraints on the distance between the endpoints of their projections. The main tool we use to deal with metric information in the cmRCD-calculus is STP. More precisely, we use STP to elaborate information on the endpoints of MBR projections onto the Cartesian axes.

Integrating the convex RCD-calculus with STP makes it possible to express both directional constraints and metric constraints in a uniform framework. As an example, the resulting formalism allows one to constrain the position of a rectangle in the plane and to impose minimum and/or maximum values to the width/height of a given rectangle, or on the vertical/horizontal distances between the sides of two rectangles. Obviously, RCD-constraints and STP-constraints are not totally independent, that is, RCD-constraints entail some metric constraints and vice versa.

Example 2. Let a and b be two rectangles. We can use the metric constraint $0 \leq a_x^+ - a_x^- \leq 7$ to state that the maximum width of a is 7 and, similarly, we can exploit the metric constraint $2 \leq a_y^+ - a_y^-$ to state that the minimum height of a is 2 (leaving the maximum height unbounded). We can also express distance constraints between the boundaries of a and b. We can constrain the horizontal distance between the right side of a and the left side of b to be at least 3 by means of the constraint $3 \leq b_x^- - a_x^+$, and the vertical distance between the upper side of a and the bottom side of b to be greater than or equal to 0 by means of the constraint $0 \leq b_y^- - a_y^+$. The two constraints together entail the basic RCD constraint $a\,\{SW\}b$. Finally, some metric constraints can be entailed by RCD ones. For instance, the convex relation $a\,\{NW, N, NE, NW{:}N, NW{:}N{:}NE, N{:}NE\}\,b$ implies that $0 \leq a_y^- - b_y^+$.

If we allow one to combine arbitrary RCD-constraints with metric constraints, then checking the consistency of the resulting set of constraints turns out to be an NP-complete problem (the consistency problem for RCD-networks is already NP-complete). To preserve tractability, the cmRCD-calculus combines convex RCD-constraints with STP-constraints. Constraint networks in the cmRCD-calculus (*cmRCD-networks*) are defined as follows. Given a convex RCD-network $N_c = (V, C)$, we denote the sets of interval variables belonging to the projections $\pi_x(toRA(N_c))$ and $\pi_y(toRA(N_c))$ by V_x and V_y, respectively. Moreover, we denote by $P(V_x)$ and $P(V_y)$ the sets of point variables representing the endpoints of the interval variables in V_x and V_y, respectively.

Definition 2. *A* cmRCD-network *is an integrated qualitative and metric constraint network N consisting of three sub-networks $(N_c, \mathcal{S}_x, \mathcal{S}_y)$, where $N_c = (V, C)$ is a convex RCD-network, and $\mathcal{S}_x = \big(P(V_x), M_x\big)$ and $\mathcal{S}_y = \big(P(V_y), M_y\big)$ are two STPs.*

Algorithm 4.1. The algorithm $\mathbf{con\text{-}cmRCD}$

```
Require: a cmRCD-network $N = (N_c, \mathcal{S}_x, \mathcal{S}_y)$
1: $N_r \leftarrow toRA(N_c)$;
2: $N_x \leftarrow \pi_x(N_r)$, $N_y \leftarrow \pi_y(N_r)$;
3: $N_x^P \leftarrow toPA(N_x)$, $N_y^P \leftarrow toPA(N_y)$;
4: $xSTP \leftarrow intersect(toSTP(N_x^P), \mathcal{S}_x)$;
5: $ySTP \leftarrow intersect(toSTP(N_y^P), \mathcal{S}_y)$;
6: if $xSTP$ or $ySTP$ is empty, then return ‘inconsistent’;
7: $xSTP^{min} \leftarrow PC_{stp}(xSTP)$;
8: $ySTP^{min} \leftarrow PC_{stp}(ySTP)$;
9: If $xSTP^{min}$ or $ySTP^{min}$ is empty, then return ‘inconsistent’; otherwise, return ‘consistent’.
```

The cmRCD-calculus subsumes both the convex RCD-calculus and the STP formalism. Moreover, it generalizes the convex fragment of the RA, since convex RA-relations are expressible as convex PA-relations, which can be encoded into an STP.

In the following, we describe an algorithm, called $\mathbf{con\text{-}cmRCD}$, to solve the consistency problem for cmRCD-networks, that runs in $O(n^3)$. As a matter of fact, a similar combination of qualitative and quantitative networks is provided by preconvex-augmented rectangle networks. An $O(n^5)$ algorithm for checking the consistency of these networks, that subsume *cmRCD-networks*, is given in [6]. We exploit the trade-off between expressiveness and complexity to obtain a more efficient consistency checking algorithm for *cmRCD-networks*.

As a preliminary step, we extend the translation mapping $toSTP$ of Table 1 to encode a convex PA-network N^P into an STP S by replacing each relation R in the network N^P by $toSTP(R)$. First, $\mathbf{con\text{-}cmRCD}$ applies the mapping $toRA$ to the input convex RCD-network N_c to get the corresponding convex RA-network N_r. Then, it computes the projections N_x and N_y of N_r. Next, it applies the mapping $toPA$ to translate the convex IA-networks N_x and N_y into two equivalent PA-networks N_x^P and N_y^P with convex relations between points variables representing the projections of the intervals in N_x and N_y. Thereafter, making use of such an encoding of convex RCD-relations as PA-relations, it looks for possible inconsistencies between these constraints and the STP-constraints on the same variables given in S_x and S_y that can be detected at this stage. To this end, it translates the PA-network N_x^P (resp., N_y^P) into an STP-network by applying the extended function $toSTP$, and then it uses the function $intersect$ to compute the “intersection” between $toSTP(N_x^P)$ and S_x (resp., $toSTP(N_y^P)$ and S_y). This function simply intersects the constraints associated with the same pairs of variables in the two STPs. If an interval intersection produces an empty interval, then $intersect$ returns an empty network, and we can conclude that N is inconsistent. Otherwise, we apply the path-consistency algorithm to the two STPs computed at lines 4 and 5 independently. The following theorem proves that $\mathbf{con\text{-}cmRCD}$ is sound and complete.

Theorem 4. *Given a cmRCD-network $N = (N_c, \mathcal{S}_x, \mathcal{S}_y)$, the algorithm* $\mathbf{con\text{-}cmRCD}$ *returns 'consistent' if and only if N is consistent.*

Proof. We basically follow the steps of the algorithm. By Theorem 2, N_c is consistent if and only if N_r is consistent, and, by Theorem 3, N_r is consistent if and only if N_x and N_y are consistent (they can be checked independently). Next, N_x and N_y are consistent if and only if N_x^P and N_y^P are consistent, since there is no loss in information in the translations [24]. The consistency of N_x^P and N_y^P can be checked by computing the corresponding STPs and by applying PC_{stp}. However, we cannot apply PC_{stp} directly to the STPs $toSTP(N_x^P)$ and $toSTP(N_x^P)$ since the metric constraints of S_x and S_y must be taken into account. Hence, we compute $xSTP = intersect(toSTP(N_x^P), S_x)$ and $ySTP = intersect(toSTP(N_x^P), S_y)$. If one of them returns an empty network, then N is inconsistent. Otherwise, we independently apply PC_{stp} to $xSTP$ and $ySTP$. By Theorem 1, if one of the two applications of PC_{stp} returns an empty network, then N is inconsistent; otherwise, the path-consistent STPs $xSTP^{min}$ and $ySTP^{min}$ are consistent (and minimal), and thus N is consistent. □

Theorem 5. *The complexity of the algorithm* $\mathbf{con\text{-}cmRCD}$ *is $O(n^3)$, where n is the number of variables of the input network.*

Proof. The translation via $toRA$, the generation of a projection of a network, the transformation of an IA-network into an RA-network via $toPA$, and the last two encodings via $toSTP$ require $O(n^2)$ steps, since there are $O(n^2)$ constraints and each constraint can be translated in constant time. The function $toPA$ introduces two variables for each interval variable, so $xSTP$ and $ySTP$ have $O(n)$ variables each. Finally, PC_{stp} runs in $O(n^3)$ time, so the overall complexity is $O(n^3)$ time (for further details about the complexity of achieving path-consistency for combined networks see [16]). □

Once we have computed the path-consistent STPs $xSTP^{min}$ and $ySTP^{min}$ with algorithm $\mathbf{con\text{-}cmRCD}$, we can build a *solution* to the cmRCD-network N by computing a solution for the points in $xSTP^{min}$ and $ySTP^{min}$, since the assignment for point variables defines a consistent assignment for rectangle variables (see Example 1). To this end, the $O(n^3)$ algorithm STP-SOLUTION, by Gerevini and Cristani [9], can be used.

5 A Case Study for the cmRCD-Calculus

Given its distinctive features, the cmRCD-calculus is well-suited for all application areas where minimal bounding rectangles can be successfully exploited. This is the case, for instance, with spatial databases [10,21], information extraction from formatted documents [8,20], and 2D-layout design [3,4,17]. We would also like to mention the application of convex RCD-calculus (the qualitative fragment of cmRCD-calculus) to the problem querying and extracting data from web documents reported in [20]. We believe that, in view of its well-balanced combination of efficiency and expressiveness, cmRCD-calculus can be naturally and successfully applied to this class of problems as well.

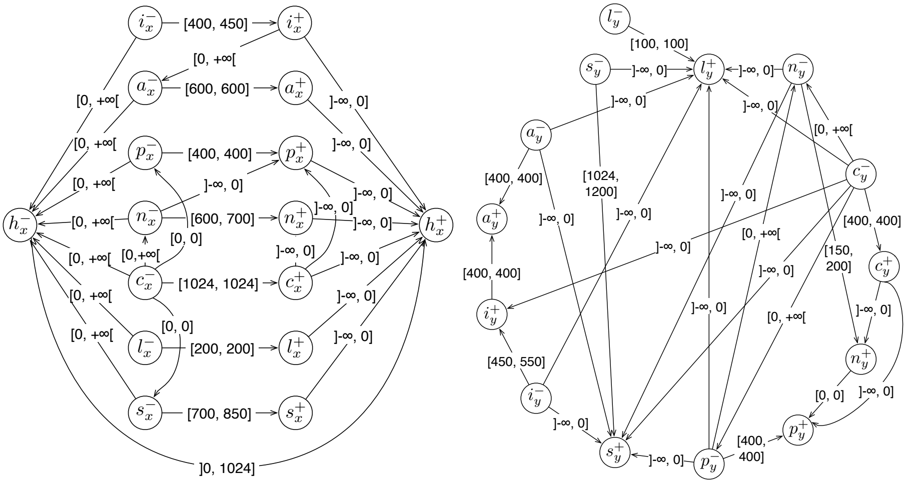


Fig. 4. Graph representation of part of the $xSTP$ and $ySTP$ of Example 3. For the sake of readability, constraints involving h in the $ySTP$ and the universal constraint are omitted.

In this section, we focus our attention on an application of the cmRCD-calculus to the automatic web page layout generation, inspired by a work by Borning et al. [4]. On the one hand, current web authoring tools do not allow home page designers to specify how the document should change in response to viewer's needs; on the other hand, web browsers do not really allow their users to express their requirements about the layout, except for those about the dimension of the font and few other features. The work by Borning et al. aims at allowing both the designer and the viewer to specify the positioning of the document elements by means of linear equalities and inequalities over their minimum bounding boxes, in such a way that the layout of the web page becomes the result of a negotiation between them (designer and viewer). In the following, we show how to apply cmRCD-calculus to allow the user (author or viewer) to specify both cardinal and metric constraints on the layout elements. Notice that, in doing that, we reduce the expressive power of Borning et al.'s proposal; nevertheless, the problem they consider is in fact an optimization problem, which is solved by means of a linear-programming algorithm that has an exponential worst-case time complexity.

Example 3. Let us consider a Facebook-like social network, that allows the user to personalize the contents of his/her home page by making use of directional constraints. We can assume each element the user can add to be represented by a MBR, or box, whose sides are parallel to the axes of the reference system centered at the lower left vertex of the home page. As an example, we may have a box containing information about his/her (gender, birthday, etc.), a box containing his/her profile picture, and so on. In addition, it makes sense to assume that the system requires all user pages to share some common presentation features like, for instance, the system logo and some general presentation directives.

In this scenario, we can imagine that a user enters the following requirements to some design tool that interacts with the layout designer (user's specifications are given in terms of directions such as “right” or “inside”, which are far more natural for a local space than their equivalent cardinal directions “to the East of” or “in the bounds of”):

1. my cover picture c has to lie on the top of the home page h, that is, the vertical distance between the top sides of c and h must be 0, and the dimension of c is 1024×400 px;
2. the box n containing my full name has to lie inside c;
3. the box i containing my personal information has to be somewhere below c, no matter what its horizontal position is (in terms of cardinal relations, this requirement should be understood as “somewhere between the south west and the south east zone of c”);
4. the box a containing the cover pictures of my photo albums has to lie to the right of i, no matter what its vertical position is; in addition, the vertical distance between the top sides of a and i must be 0;
5. the box p containing my profile picture has to lie inside c, and the horizontal distance between the left sides of of p and c must be 0; in addition, n has to be to the right of p, with the restriction that the vertical distance between the top sides of p and n must be 0;
6. I want to see the 5 most recent stories from my friends in a box s, which has to lie below the previous elements, no matter what its horizontal position is, and the horizontal distance between the left sides of s and c must be 0.

Besides these user constraints, we can imagine that the system imposes the following additional constraints:

7. somewhere at the bottom of the home page, there must be a logo l, whose dimension is 200×100 px;
8. the width of the home page cannot exceed 1024 px.

When the contents of all boxes are retrieved from the database server, the system provides lower and upper bounds to the size of the boxes, so that the layout manager has more chances to fit the contents on the basis of user preferences. In particular, we can assume that the following conditions hold:

9. n’s width can vary from 600 to 700 px, while n’s height can vary from 150 to 200 px;
10. i’s width can vary from 400 to 450 px, while i’s height can vary from 450 to 550 px;
11. s’s width can vary from 700 to 850 px, while s’s height can vary from 1024 to 1200 px;
12. a is 600×400 px;
13. p is 400×400 px.

A particular choice for the size of a box is automatically compensated by increasing/decreasing the font size.

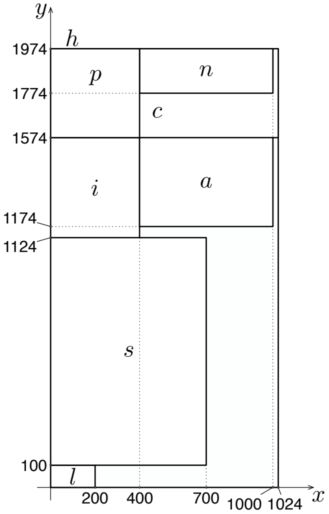


Fig. 5. A solution to the cmRCD-network for Example 3.

When the system has to deliver the web page, it must find a solution to a *cmRCD-network* consisting of the following qualitative constraints (that encode the above qualitative requirements):

1. Implicit: “boxes must be inside the homepage”:
   $$c\,B\,h,\quad n\,B\,h,\quad i\,B\,h,\quad a\,B\,h,\quad l\,B\,h,\quad p\,B\,h,\quad s\,B\,h$$
2. $n\,B\,c$;
3. $i\,\{SW, S, SW{:}S, SW{:}S{:}SE, S{:}SE, SE\}\,c$;
4. $a\,\{NE, E, NE{:}E, NE{:}E{:}SE, E{:}SE, SE\}\,i$;
5. $p\,B\,c,\ n\,E\,p$;
6. for each box $b \in \{c, n, i, a, p\}$, $s\,\{SW, S, SW{:}S, SW{:}S{:}SE, S{:}SE, SE\}\,b$;
7. for each box b, $l\,\{SW, S, SW{:}S, SW{:}S{:}SE, S{:}SE, SE\}\,b$;

and of the following metric constraints (that encode the above metric requirements):

1. $h_y^+ - c_y^+ = 0,\ c_y^+ - c_y^- = 400,\ c_x^+ - c_x^- = 1024$;
2. $a_y^+ - i_y^+ = 0$;
3. $p_y^+ - n_y^+ = 0,\ p_x^- - c_x^- = 0$;
4. $s_x^- - c_x^- = 0$;
5. $l_x^+ - l_x^- = 200,\ l_y^+ - l_y^- = 100$;
6. $0 < h_x^+ - h_x^- \leq 1024$;
7. $600 \leq n_x^+ - n_x^- \leq 700,\ 150 \leq n_y^+ - n_y^- \leq 200$;
8. $400 \leq i_x^+ - i_x^- \leq 450,\ 450 \leq i_y^+ - i_y^- \leq 550$;
9. $700 \leq s_x^+ - s_x^- \leq 850,\ 1024 \leq s_y^+ - s_y^- \leq 1200$;
10. $a_x^+ - a_x^- = 600,\ a_y^+ - a_y^- = 400$;
11. $p_x^+ - p_x^- = 400,\ p_y^+ - p_y^- = 400$.

Meaningful portions of the constraint networks $xSTP$ and $ySTP$, generated by steps 4 and 5 of the algorithm **con-cmRCD**, respectively, are depicted in Figure 4. A possible solution to the network, that is, a possible layout of the user home page, is given in Figure 5. It is worth pointing out that the picture shows the minimum feasible values for point variables, and thus it immediately follows that the minimum dimension of h is 1024×1974 px.

6 Conclusions

In this paper, we have proposed a quite expressive, but tractable, metric extension of RCD-calculus, that integrates convex RCD-constraints and STP-constraints. The resulting cmRCD-calculus allows one to constrain the position of a rectangle in the plane, its width/height, and the vertical/horizontal distance between the sides of two rectangles, as well as to represent cardinal relations between rectangles. We have developed an $O(n^3)$ consistency-checking algorithm for such a calculus, and we have shown how a spatial realization of a *cmRCD-network* can be built.

As for future work, we plan to extend the cmRCD-calculus with topological relations to improve its expressiveness (similar results can be found in [11,13]). Moreover, since the problem of identifying maximal tractable subsets of RCD is still open, it makes sense to search for tractable classes strictly including the convex fragment. Finally, we are interested in developing heuristics and algorithms to check consistency and to find a solution in the cases of non-convex RCD-relations or disjunctive metric constraints. Since in these cases both problems turn out to be intractable, an empirical evaluation of the solutions is necessary to check scalability.

Acknowledgements. This work has been partially supported by the Spanish Ministry of Science and Innovation, the European Regional Development Fund of the European Commission under grant TIN2009-14372-C03-01, and the Spanish MEC through the project 15277/PI/10, funded by Seneca Agency of Science and Technology of the Region of Murcia within the II PCTRM 2007-2010. Guido Sciavicco and Angelo Montanari were also partially founded by the Spanish fellowship 'Ramon y Cajal' RYC-2011-07821 and by the Italian PRIN project *Innovative and multi-disciplinary approaches for constraint and preference reasoning*, respectively.

References

1. Allen, J.F.: Maintaining knowledge about temporal intervals. Communications of the ACM 26(11), 832–843 (1983)
2. Balbiani, P., Condotta, J.F., del Cerro, L.F.: A model for reasoning about bidimensional temporal relations. In: Proc. of KR 1998, pp. 124–130 (1998)
3. Baykan, C.A., Fox, M.D.: Spatial synthesis by disjunctive constraint satisfaction. Artificial Intelligence for Engineering Design, Analysis and Manufacturing 11(4), 245–262 (1997)
4. Borning, A., Kuang-Hsu Lin, R., Marriott, K.: Constraint-based document layout for the web. Multimedia Syst. 8(3), 177–189 (2000)
5. Cohn, A.G., Hazarika, S.M.: Qualitative spatial representation and reasoning: An overview. Fundamenta Informaticae 46(1-2), 1–29 (2001)
6. Condotta, J.F.: The augmented interval and rectangle networks. In: Proc. of KR 2000, pp. 571–579 (2000)

7. Dechter, R., Meiri, I., Pearl, J.: Temporal constraint networks. Artificial Intelligence 49(1-3), 61–95 (1991)
8. Gatterbauer, W., Bohunsky, P.: Table extraction using spatial reasoning on the CSS2 visual box model. In: Proc. of AAAI 2006, pp. 1313–1318 (2006)
9. Gerevini, A., Cristani, M.: On finding a solution in temporal constraint satisfaction problems. In: Proc. of IJCAI 1997, vol. 2, pp. 1460–1465 (1997)
10. Goyal, R., Egenhofer, M.: Consistent queries over cardinal directions across different levels of detail. In: Proc. of DEXA 2000, pp. 876–880 (2000)
11. Gerevini, A., Renz, J.: Combining topological and size information for spatial reasoning. Artificial Intelligence 137(1-2), 1–42 (2002)
12. Liu, W., Li, S.: Reasoning about cardinal directions between extended objects: The NP-hardness result. Artificial Intelligence 175(18), 2155–2169 (2011)
13. Liu, W., Li, S., Renz, J.: Combining RCC-8 with qualitative direction calculi: Algorithms and complexity. In: Proc. of IJCAI 2009, pp. 854–859 (2009)
14. Liu, W., Zhang, X., Li, S., Ying, M.: Reasoning about cardinal directions between extended objects. Artificial Intelligence 174, 951–983 (2010)
15. Mackworth, A.K.: Consistency in Networks of Relations. Artificial Intelligence 8(1), 99–118 (1977)
16. Meiri, I.: Combining qualitative and quantitative constraints in temporal reasoning. Artificial Intelligence 87(1-2), 343–385 (1996)
17. Montanari, A., Navarrete, I., Sciavicco, G., Tonon, A.: A tractable formalism for combining rectangular cardinal relations with metric constraints. In: Proc. of ICAART 2012, pp. 154–163 (2012)
18. Navarrete, I., Morales, A., Sciavicco, G., Cardenas, M.A.: Spatial reasoning with rectangular cardinal relations utility package for rectangular cardinal relations. Technical Report TR-DIIC 2/11, Universidad de Murcia (2011)
19. Navarrete, I., Sciavicco, G.: Spatial reasoning with rectangular cardinal direction relations. In: Proc. of the ECAI 2006 Workshop on Spatial and Temporal Reasoning, pp. 1–10 (2006)
20. Oro, E., Ruffolo, M., Staab, S.: SXPath - extending XPath towards spatial querying on web documents. Proc. of VLDB 2010 4(2), 129–140 (2010)
21. Papadias, D., Theodoridis, Y.: Spatial relations, minimum bounding rectangles, and spatial data structures. International Journal of Geographical Information Science 11(2), 111–138 (1997)
22. Skiadopoulos, S., Koubarakis, M.: On the consistency of cardinal directions constraints. Artificial Intelligence 163(1), 91–135 (2005)
23. van Beek, P.: Reasoning about qualitative temporal information. Artificial Intelligence 58(1-3), 297–326 (1992)
24. van Beek, P., Cohen, R.: Exact and approximate reasoning about temporal relations. Computation Intelligence 6(3), 132–147 (1990)
25. Vilain, M.B., Kautz, H.: Constraint propagation algorithms for temporal reasoning. In: Proc. of AAAI 1986, pp. 377–382 (1986)

Sorting High-Dimensional Patterns with Unsupervised Nearest Neighbors

Oliver Kramer

Department of Computer Science, University of Oldenburg,
Uhlhornsweg 84, 26111 Oldenburg, Germany
oliver.kramer@uni-oldenburg.de

Abstract. In many scientific disciplines structures in high-dimensional data have to be detected, e.g., in stellar spectra, genome data, or in face recognition tasks. In this work we present an approach to non-linear dimensionality reduction based on fitting nearest neighbor regression to the unsupervised regression framework for learning low-dimensional manifolds. The problem of optimizing latent neighborhoods is difficult to solve, but the unsupervised nearest neighbor (UNN) formulation allows an efficient strategy of iteratively embedding latent points to discrete neighborhood topologies. The choice of an appropriate loss function is relevant, in particular for noisy, and high-dimensional data spaces. We extend UNN by the ϵ-insensitive loss, which allows to ignore small residuals under a defined threshold. Furthermore, we introduce techniques to handle incomplete data. Experimental analyses on various artificial and real-world test problems demonstrates the performance of the approaches.

Keywords: Dimensionality reduction, Unsupervised regression, Nearest neighbors, Robust loss functions, Missing data.

1 Introduction

Dimensionality reduction and manifold learning have an important part to play in the understanding of data. Many disciplines in science and economy are based on collecting high-dimensional patterns: from astronomy to psychology, from civil engineering to social web services. Algorithms are required that are able to process data efficiently. The collection and understanding of data allows us to improve the efficiency of processes in a variety of domains. There are numerous examples that reflect the importance of the understanding of large data sets. The quality of sensors is steadily being improved. The trend towards digitizing the world leads to large amounts of high-dimensional patterns. For an efficient data analysis process fast dimensionality reduction methods are required. UNN is a fast iterative approach based on unsupervised regression. The idea of unsupervised regression is to reverse functional regression models such that low-dimensional data samples in latent space optimally reconstruct high-dimensional output data. We take this framework as basis for an iterative approach that fits K-nearest neighbors (KNN) regression into this unsupervised setting.

The manifold problem we consider is a point-wise mapping $\mathbf{F} : \mathbf{y} \rightarrow \mathbf{x}$ from patterns $\mathbf{y} \in \mathbb{R}^d$ to latent embeddings $\mathbf{x} \in \mathbb{R}^q$ with $d > q$. The problem is a hard optimization problem as the latent variables $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ are unknown.

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 250–267, 2013.

In Section 2 we will review related work in dimensionality reduction, unsupervised regression, and KNN regression. Section 3 presents the concept of UNN regression, and two iterative strategies that are based on fixed latent space topologies. In Section 4 we extend UNN to robust loss functions, i.e., the ϵ-insensitive loss. In Section 5 we will show how the constructive variants can be extended to handle incomplete data. Conclusions are drawn in Section 6.

2 Related Work

Dimensionality reduction is the problem of learning a mapping from high-dimensional data space to a space with lower dimensions, while losing as little information as possible. Many dimensionality reduction methods have been proposed in the past, a very famous one is principal component analysis (PCA), which assumes linearity of the manifold [14,22]. An extension for learning non-linear manifolds is kernel PCA [25] that projects the data into a Hilbert space. Further famous approaches for dimensionality reduction are ISOMAP by Tenenbaum *et al.* [28], locally linear embedding (LLE) by Roweis and Saul [23], and principal curves by Hastie and Stuetzle [12]. An introduction to other dimensionality reduction methods can be found in machine learning textbooks like [4], and [11].

2.1 Unsupervised Regression

The work on unsupervised regression for dimensionality reduction started with Meinicke [20], who introduced the corresponding algorithmic framework for the first time. In this line of research early work concentrated on non-parametric kernel density regression, i.e., the counterpart of the Nadaraya-Watson estimator denoted as unsupervised kernel regression (UKR) [21].

Unsupervised regression works as follows. Let $\mathbf{Y} = (\mathbf{y}_1, \ldots \mathbf{y}_N)$ with $\mathbf{y}_i \in \mathrm{I\!R}^d$ be the matrix of high-dimensional patterns in data space. We seek for a low-dimensional representation, i.e., a matrix of latent points $\mathbf{X} = (\mathbf{x}_1, \ldots \mathbf{x}_N)$, so that a regression function $\mathbf{f}$ applied to $\mathbf{X}$ *point-wise optimally reconstructs the patterns*, i.e., we search for an $\mathbf{X}$ that minimizes the reconstruction in data space. The optimization problem can be formalized as follows:

$$\text{minimize } E(\mathbf{X}) = \frac{1}{N} \|\mathbf{Y} - \mathbf{f}(\mathbf{x}; \mathbf{X})\|^2. \tag{1}$$

$E(\mathbf{X})$ is called data space reconstruction error (DSRE). Latent points $\mathbf{X}$ define the low-dimensional representation. The regression function $\mathbf{f}$ applied to the latent points should optimally *reconstruct* the high-dimensional patterns. The regression model $\mathbf{f}$ induces its characteristics to the mapping.

Most unsupervised regression approaches are based on the iterative improvement of a spectral embedding solution, and work as follows:

- Initialize latent variables $\mathbf{X} = (\mathbf{x}_1, \ldots \mathbf{x}_N)$ (often employing spectral methods like LLE), and
- minimize $E(\mathbf{X})$ w.r.t. DSRE employing an optimization scheme.

A typical example is unsupervised kernel regression, analyzed by Klanke and Ritter [16], but further methods can also be employed. Klanke and Ritter [16] introduced an optimization scheme based on LLE, PCA, and leave-one-out cross-validation (LOO-CV) for UKR. Various extension of UKR have been proposed, e.g., a feature space variant (employing Mercer kernels), and the use of landmark points to reduce the non-parametric training set. Carreira-Perpiñán and Lu [5] argue that training of non-parametric unsupervised regression approaches is quite expensive, i.e., $\mathcal{O}(N^3)$ in time, and $\mathcal{O}(N^2)$ in memory. Parametric methods can accelerate learning, e.g., unsupervised regression based on radial basis function networks (RBFs) [26], Gaussian processes [19], and neural networks [27].

2.2 Nearest Neighbor Regression

In the following, we give a short introduction to K-nearest neighbor regression that is basis of the UNN approach. KNN is a technique with a long tradition. It was first mentioned by Fix and Hodges [8] in the fifties in an unpublished US Air Force School of Aviation Medicine report as non-parametric classification technique. Cover and Hart [3] investigated the approach experimentally in the sixties. Interesting properties have been found, e.g., that for $K = 1$, and $N \rightarrow \infty$, KNN is bound by the Bayes error rate.

The problem in regression is to predict output values $\mathbf{y} \in \mathbb{R}^d$ of given input values $\mathbf{x} \in \mathbb{R}^q$ based on sets of N input-output examples $((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N))$. The goal is to learn a function $\mathbf{f} : \mathbf{x} \rightarrow \mathbf{y}$ known as regression function. For a novel pattern $\mathbf{x}'$ KNN regression computes the mean of the function values of its K-nearest neighbors:

$$\mathbf{f}_{KNN}(\mathbf{x}') := \frac{1}{K} \sum_{i \in \mathcal{N}_K(\mathbf{x}')} \mathbf{y}_i \tag{2}$$

with set $\mathcal{N}_K(\mathbf{x}')$ containing the indices of the K-nearest neighbors of $\mathbf{x}'$. The idea of KNN is based on the assumption of locality in data space: In local neighborhoods of a pattern $\mathbf{x}$ near patterns are expected to have similar labels $\mathbf{y}$. Consequently, for an unknown $\mathbf{x}'$ the label must be similar to the labels of the closest patterns, which is modeled by the average of the output value of the K nearest samples. KNN has been proven well in various applications, e.g., in the detection of quasars based on spectroscopic data [10].

3 Unsupervised Nearest Neighbor Regression

In this section we introduce an iterative strategy for UNN regression [17,18] that is based on minimization of the data space reconstruction error (DSRE).

3.1 UNN Regression Problem

An UNN regression manifold is defined by variables $\mathbf{x}$ from $\mathbf{X} \in \mathrm{I\!R}^{q \times N}$ with unsupervised formulation of an UNN regression manifold

$$\mathbf{f}_{UNN}(\mathbf{x}; \mathbf{X}) := \frac{1}{K} \sum_{i \in \mathcal{N}_K(\mathbf{x}, \mathbf{X})} \mathbf{y}_i. \tag{3}$$

Matrix $\mathbf{X}$ contains the latent points $\mathbf{x}$ that define the manifold, i.e., the low-dimensional representation of patterns $\mathbf{Y}$. Parameter $\mathbf{x}$ is the location where the function is evaluated. An optimal UNN regression manifold minimizes the DSRE:

$$\text{minimize } E(\mathbf{X}) = \frac{1}{N} \|\mathbf{Y} - \mathbf{f}_{UNN}(\mathbf{x}; \mathbf{X})\|_F^2, \tag{4}$$

with Frobenius norm $\|\mathbf{A}\|_F := \sqrt{\sum_{i=1}^{d} \sum_{j=1}^{N} |a_{ij}|^2}$. In other words: an optimal UNN manifold consists of low-dimensional points $\mathbf{X}^*$ that minimize the reconstruction of the data points $\mathbf{Y}$ w.r.t. KNN regression in data space. Regularization for UNN regression is not as important as regularization for other methods that fit into the unsupervised regression framework. For example, in UKR regularization means penalizing extension in latent space with $E_p(\mathbf{X}) = E(\mathbf{X}) + \lambda\|\mathbf{X}\|$, and weight λ [16]. In KNN regression moving the low-dimensional data samples infinitely apart from each other does not have the same effect as long as we can still determine the K-nearest neighbors. For the fixed discrete latent positions regularization is not necessary as the latent points cannot move arbitrarily in latent space.

3.2 Iterative Strategy

For KNN not the absolute positions of data samples in latent space are relevant, but the relative positions that define the *neighborhood relations*. This perspective reduces the problem to a combinatorial search for neighborhoods $\mathcal{N}_K(\mathbf{x}_i, \mathbf{X})$ with $i = 1, \ldots, N$ that can be solved by testing all combinations of K-element subsets of N elements. The problem is still difficult to solve, in particular for high dimensions.

The idea of the basic iterative UNN strategy is to iteratively assign the data samples to a position in an existing latent space topology that leads to the lowest DSRE. We assume fixed neighborhood topologies with equidistant positions in latent space, and therefore restrict the optimization problem of Equation (3) to a search on a discrete latent topology.

As a simple variant we consider the linear case $q = 1$ with latent variables arranged equidistantly on a line, i.e. $\mathbf{x} \in \mathrm{I\!R}$. In this simplified case only the order of the elements is important. The first iterative strategy works as follows:

1. Choose a pattern $\mathbf{y}$ from matrix $\mathbf{Y}$,
2. test all $\hat{N} + 1$ intermediate positions of the $\hat{N} = |\hat{\mathbf{Y}}|$ embedded elements $\hat{\mathbf{Y}}$ in latent space,

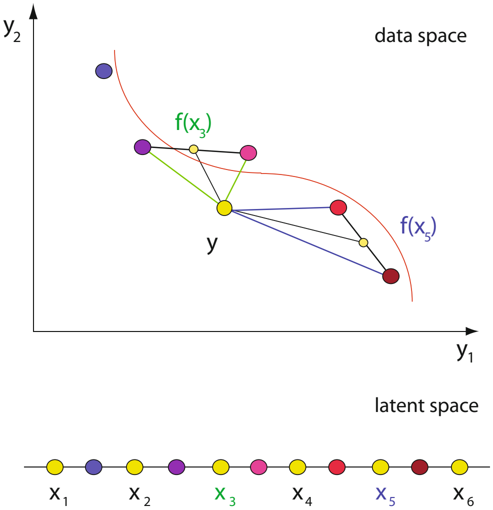


Fig. 1. UNN: illustration of embedding of a low-dimensional point to a discrete latent space topology w.r.t. the DSRE testing all $\hat{N}+1$ intermediate positions

3. choose the latent position that minimizes $E(\mathbf{X})$, and embed $\mathbf{y}$,
4. remove $\mathbf{y}$ from $\mathbf{Y}$, add $\mathbf{y}$ to $\hat{\mathbf{Y}}$, and repeat from Step 1 until all patterns have been embedded ($\hat{N}=0$).

Figure 1 illustrates the $\hat{N}+1$ possible embeddings of a pattern into an existing order of points in latent space (yellow/bright circles). The position of element $\mathbf{x}_3$ results in a lower DSRE with $K=2$ than the position of $\mathbf{x}_5$, as the mean of the two nearest neighbors of $\mathbf{x}_3$ is closer to $\mathbf{y}$ than the mean of the two nearest neighbors of $\mathbf{x}_5$.

3.3 Greedy Strategy

The iterative approach introduced in the last section tests all intermediate positions of previously embedded latent points. We proposed a second iterative variant (UNN$_g$) that only tests the neighbored intermediate positions in latent space of the nearest embedded point $\mathbf{y}^*$ from $\hat{\mathbf{Y}}$ in data space [17]. The second iterative strategy works as follows:

1. Choose a pattern $\mathbf{y}$ from $\mathbf{Y}$,
2. look for the nearest $\mathbf{y}^*$ from $\hat{\mathbf{Y}}$ that has already been embedded (w.r.t. distance measure like Euclidean distance),
3. choose the latent position next to $\mathbf{x}^*$ that minimizes $E(\mathbf{X})$ and embed $\mathbf{y}$,
4. remove $\mathbf{y}$ from $\mathbf{Y}$, add $\mathbf{y}$ to $\hat{\mathbf{Y}}$, and repeat from Step 1 until all patterns have been embedded.

Figure 2 illustrates the embedding of a 2-dimensional point $\mathbf{y}$ (yellow) left or right of the nearest point $\mathbf{y}^*$ in data space. The position with the lowest DSRE is chosen. In comparison to UNN, $\hat{N}$ distance comparisons in data space have to be computed, but only two positions have to be tested w.r.t. the data space reconstruction error.

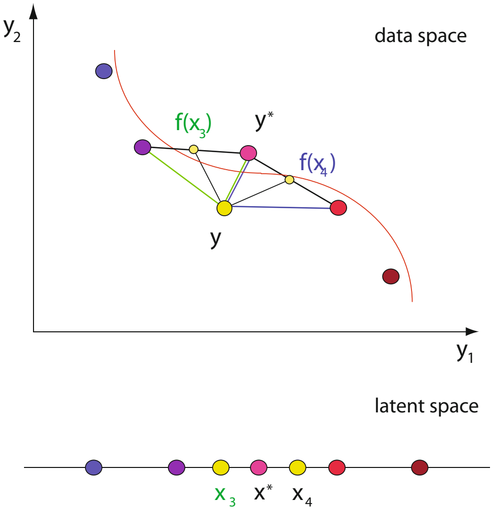
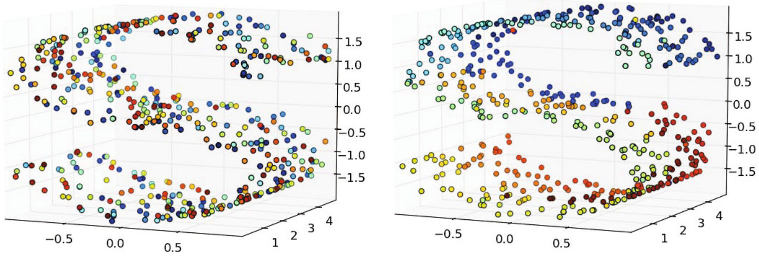


Fig. 2. UNN$_g$: testing only the neighbored positions of the nearest point $\mathbf{y}^*$ in data space



(a) unsorted S (b) UNN embedding of S

Fig. 3. Results of UNN on 3D-S: (a) the unsorted S at the beginning, (b) the embedded S with UNN and $K = 10$. Similar colors represent neighborhood relations in latent space.

3.4 Experiments

In the following, we present an experimental evaluation of UNN regression on artificial test data (2-, and 3-dimensional S data set, USPS digits data set [13]), and real-world data from astronomy.

***S* and USPS.** The 3D-S variant without a hole (3D-S$_h$) consists of 500 data points. Figure 3 (a) shows the order of elements of the 3D-S data set at the beginning. The corresponding embedding with UNN and $K = 10$ is shown in Figure 3 (b). Similar colors correspond to neighbored points in latent space. Figure 4 shows the embedding of 100 data samples of 256-dimensional (16 x 16 pixels) images of handwritten digits (2's). We embed a one-dimensional manifold, and show the high-dimensional data that is assigned to every 14th latent point. We can observe that neighbored digits are similar

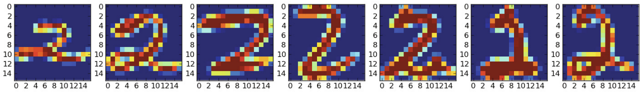


Fig. 4. UNN$_g$ embeddings of 100 digits (2's) from the USPS data set. The images are shown that are assigned to every 14th embedded latent point. Similar digits are neighbored in latent space.

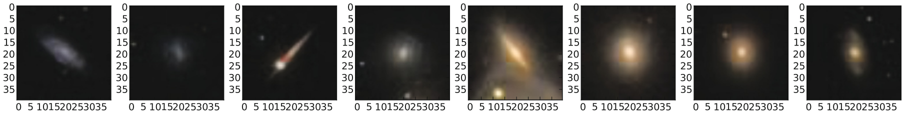


Fig. 5. Sorting of Galaxies with UNN$_g$. Galaxies from similar classes in the Hubble sequence are neighbored in latent space.

Table 1. Comparison of DSRE for initial data set, and after embedding with strategy UNN, and UNN$_g$

	2D-S			3D-S		
K	2	5	10	2	5	10
init	201.6	290.0	309.2	691.3	904.5	945.80
UNN	**19.6**	**27.1**	66.3	**101.9**	**126.7**	**263.39**
UNN$_g$	29.2	70.1	64.7	140.4	244.4	296.5
LLE	25.5	37.7	**40.6**	135.0	514.3	583.6
	3D-S$_h$			digits (7)		
K	2	5	10	2	5	10
init	577.0	727.6	810.7	196.6	248.2	265.2
UNN	**80.7**	**108.1**	**216.4**	**139.0**	**179.3**	**216.6**
UNN$_g$	101.8	204.4	346.8	145.3	195.4	222.1
LLE	94.9	198.9	387.4	147.8	198.1	217.8

to each other, while digits that are dissimilar are further away from each other in latent space.

DSRE Comparison. Besides the visual interpretation we compare the DSRE achieved by both strategies with the initial DSRE, and the DSRE achieved by LLE on all test problems. For the USPS digits data set we choose the number 7. Table 1 shows the experimental results of three settings for the neighborhood size K. The lowest DSRE on each problem is highlighted with bold figures. After application of the iterative strategies the DSRE is significantly lower than initially. Increasing K results in higher DSREs. With exception of LLE with $K = 10$ on 2D-S, the UNN strategy always achieves the best results. UNN achieves lower DSREs than UNN$_g$, with exception of 2D-S, and $K = 10$. The win in accuracy has to be paid with a constant runtime factor that may play an important role in case of large sets of high-dimensional data.

Sorting of Galaxies. In the following, we test UNN on real-world data from astronomy, i.e., images of galaxies. Galaxies are a massive, gravitationally bound systems of stars, gas, and dust. The number of stars typically varies in the range of 10^7 to 10^{14}. Edwin Hubble introduced a morphological classification scheme based, which became famous as Hubble sequence. Neighbored classes in this diagram represent galaxies with similar shape. Hubble's classification scheme differentiates between three main classes: (1) elliptical galaxies, (2) spiral galaxies, and (3) lenticular galaxies, see [15]. For our experiment we employ images of galaxies from the *Sloan Digital Sky Survey (SDSS)*, a collection of millions of astronomical objects [1]. Figure 5 shows the UNN$_g$ embedding of 100 images of galaxies from the SDSS data basis. Each image is a vector of 40×40-RGB values, i.e., the data space dimensionality is $d = 4,800$. The figure shows every 12th galaxy. We can observe that galaxies, which belong to one class according to Hubble's classification scheme are neighbored on the low-dimensional manifold. Elliptical galaxies start from the left, while lenticular shapes are places on the right hand side, a sorting that is similar to the Hubble taxonomy.

4 Robust Loss Functions

Loss functions have an important part to play in machine learning, as they define the error, and thus the design objective while training a functional model. In particular, in the presence of noise the choice of an appropriate loss function parameterization has an important part to play. In this section we extend UNN regression by the ϵ-insensitive loss.

4.1 The ϵ-Insensitive Loss

In case of noisy data sets over-fitting effects may occur. The employment of the ϵ-insensitive loss allows to ignore errors beyond a level of ϵ, and avoids over-fitting to curvatures of the data that may only be caused by noise effects. With the design of a loss function, the emphasis of outliers can be controlled. First, the residuals are computed. In case of unsupervised regression, the error is computed in two steps:

1. The distance function $\delta : \mathbb{R}^q \times \mathbb{R}^d \rightarrow \mathbb{R}$ maps the difference between the prediction $f(\mathbf{x})$ and the desired output value $\mathbf{y}$ to a value according to the distance w.r.t. a certain measure. We employ the Minkowski metric:

$$\delta(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{N} |\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)| \right)^{1/p}, \tag{5}$$

   which corresponds to the Manhattan distance for $p = 1$, and to the Euclidean distance for $p = 2$.
2. The loss function $L : \mathbb{R} \rightarrow \mathbb{R}$ maps the residuals to the learning error. With the design of the loss function the influence of residuals can be controlled. In the best case the loss function is chosen according to the requirements of the underlying data mining model. Often, low residuals are penalized less than high residuals (e.g. with a quadratic function). We will concentrate on the ϵ-insensitive loss in the following.

Let r be the residual, i.e., the distance δ in data space. $L_1(r) = \|r\|$, and $L_2 = r^2$ are often employed as loss functions. We will use the L_2 loss for measuring the final DSRE, but concentrate on the ϵ-insensitive loss L_ϵ during training of the UNN model. The L_ϵ is defined as:

$$L_\epsilon(r) = \begin{cases} 0 & \text{if } |r| < \epsilon \\ |r| - \epsilon & \text{if } |r| \geq \epsilon \end{cases} \tag{6}$$

L_ϵ is not differentiable at $|r| = \epsilon$. In contrast to the L_1 and the L_2 loss L_ϵ ignores residuals below ϵ, and thus avoids over-adaptation to noise.

4.2 Experiments

In the following, we concentrate on the influence of loss functions on the UNN learning results. For this sake, we employ two kinds of ways to evaluate the final embedding: We measure the final L_2-based DSRE, visualize the results by colored embeddings, and show the latent order of the embedded objects. Again, we concentrate on two data sets, i.e., a 3D-S data set with noise, and the USPS handwritten digits.

3D-S with Noise. In the first experiment we concentrate on the 3D-S data set. Noise is modeled by multiplying each data point of the 3D-S with a random value drawn from the Gaussian distribution: $\mathbf{y}' = \mathcal{N}(0, \sigma) \cdot \mathbf{y}$. Table 2 shows the experimental results of UNN and UNN$_g$ concentrating on the ϵ-insensitive loss for $K = 5$, and Minkowski metric with $p = 2$ on the 3D-S data set with hole (3D-S$_h$). The left part shows the results for 3D-S without noise, the right part shows the results with noise ($\sigma = 5.0$). At first, we concentrate on the experiments without noise. We can observe that (1) the DSRE achieved by UNN is minimal for the lowest ϵ, and (2), for UNN$_g$ low DSRE values are achieved with increasing ϵ (to a limit as of $\epsilon = 3.0$), but the best DSRE of UNN$_g$ is worse than the best of UNN. Observation (1) can be explained as follows. Without noise for UNN ignoring residuals is disadvantageous: all intermediate positions are tested, and a good local optimum can be reached. For observation (2) we can conclude that a good strategy against local optima of UNN$_g$ is to ignore residuals beyond threshold ϵ.

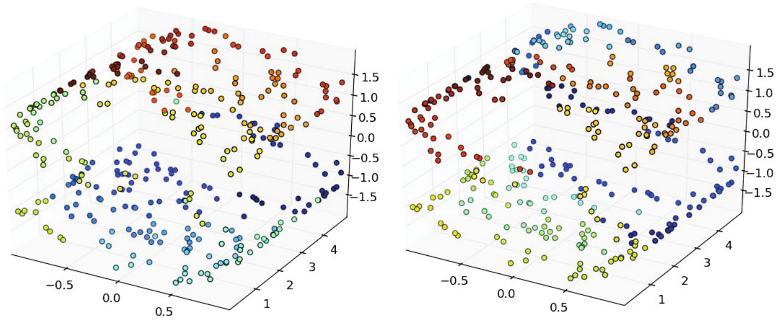
For the experiments with noise of the magnitude $\sigma = 5.0$ we can observe a local DSRE minimum: for $\epsilon = 0.8$ in case of UNN, and $\epsilon = 3.0$ in case of UNN$_g$. For UNN local optima caused by noise can be avoided by ignoring residuals, for UNN$_g$ this is already the case without noise. Furthermore, for UNN$_g$ we observe the optimum at the same level of ϵ.

Figures 6 (a) and (b) show embeddings of UNN and UNN$_g$ without noise, and the settings $\epsilon = 0.2$, and $\epsilon = 3.0$, corresponding to the settings of Table 2 that are shown in bold. Similar colors correspond to neighbored embeddings in latent space. The visualization shows that for both embeddings neighbored points in data space have similar colors, i.e., they correspond to neighbored latent points. The UNN embedding results in a lower DSRE. This can hardly be recognized from the visualization. Only the blue points of UNN$_g$ seem to be misplaced on the upper and lower part of the 3D-S.

Figures 7 (a) and (b) show the visualization of the UNN embeddings on the noisy 3D-S. The structure of the 3-dimensional S is obviously disturbed. Nevertheless, neighbored parts in data space are assigned to similar colors. Again, the UNN embedding

Table 2. Influence of the ϵ-insensitive loss on final DSRE (L_2) of UNN for problem 3D-S$_h$ with, and without noise

	$\sigma = 0.0$		$\sigma = 5.0$	
ϵ	UNN	UNN$_g$	UNN	UNN$_g$
0.2	**47.432**	77.440	79.137	85.609
0.4	48.192	77.440	79.302	85.609
0.6	51.807	76.338	78.719	85.609
0.8	50.958	76.338	**77.238**	84.422
1.0	64.074	76.427	79.486	84.258
2.0	96.026	68.371	119.642	82.054
3.0	138.491	**50.642**	163.752	**80.511**
4.0	139.168	50.642	168.898	82.144
5.0	139.168	50.642	169.024	83.209
10.0	139.168	50.642	169.024	83.209



(a) UNN, $\epsilon = 0.2$

(b) UNN$_g$, $\epsilon = 3.0$

Fig. 6. Visualization of the best UNN and UNN$_g$ embeddings (lowest DSRE, bold values in Table 2) of 3D-S$_h$ without noise

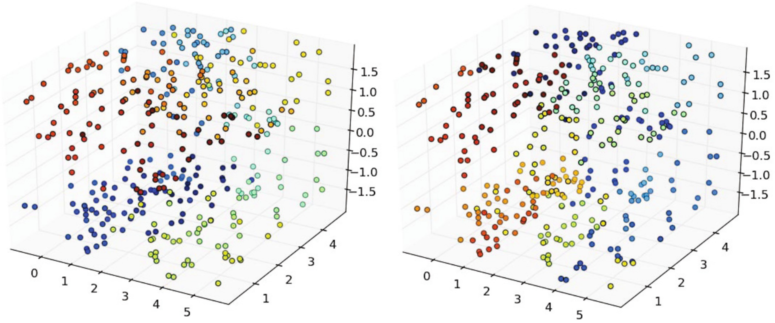
seems to be slightly better than the UNN$_g$ embedding, blue points can again be observed at different parts of the structure, representing local optima.

USPS Digits. To demonstrate the effect of the ϵ-insensitive loss for data spaces with higher dimensions, we employ the USPS handwritten digits data set with $d = 256$ again by showing the DSRE, and presenting a visualization of the embeddings. Table 3 shows the final DSRE (w.r.t. the L_2-loss) after training with the ϵ-insensitive loss with various parameterizations for ϵ. We used the setting $K = 10$, and $p = 10.0$ for the Minkowski metric. The results for digit 5 show that a minimal DSRE has been achieved for $\epsilon = 3.0$ in case of UNN, and $\epsilon = 5.0$ for UNN$_g$ (a minimum of $R = 429.75561$ was found for $\epsilon = 4.7$). Obviously, both methods can profit from the use of the ϵ-insensitive loss. For digit 7, and UNN ignoring small residuals does not seem to improve the learning result, while for UNN$_g$ $\epsilon = 4.0$ achieves the best embedding.

Figure 8 shows two UNN$_g$ embeddings of the handwritten digits data set for $\epsilon = 2.0$ (a), and $\epsilon = 20.0$ (b). For both settings similar digits are neighbored in latent space. But we can observe that for $\epsilon = 20.0$ a broader variety in the data set is covered. The loss function does not concentrate on fitting to noisy parts of the data, but has the capacity to concentrate on the important structures of the data.

Table 3. Influence of ϵ-insensitive loss on final DSRE of UNN on the digits data set

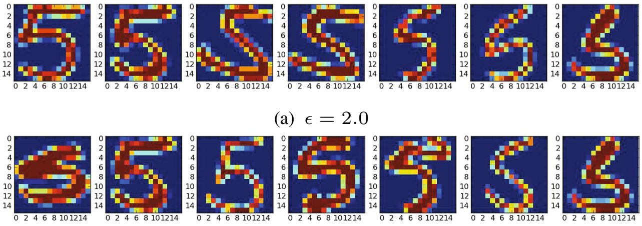
	digits 5's		digits 7's	
ϵ	UNN	UNN$_g$	UNN	UNN$_g$
0.0	423.8	440.2	**225.4**	222.8
1.0	423.8	440.2	**225.4**	222.8
2.0	423.8	440.2	225.6	222.8
3.0	**423.5**	440.2	238.1	221.0
4.0	441.3	440.2	262.1	**218.2**
5.0	488.7	**432.3**	264.8	221.4
6.0	496.9	434.2	265.6	220.8
10.0	494.6	434.3	268.4	220.8



(a) UNN, $\epsilon = 0.8$

(b) UNN$_g$, $\epsilon = 3.0$

Fig. 7. Visualization of the best UNN and UNN$_g$ embeddings (lowest DSRE, bold values in Table 2) of 3D-S$_h$ with noise $\sigma = 5.0$



(a) $\epsilon = 2.0$

(b) $\epsilon = 20.0$

Fig. 8. Comparison of UNN$_g$ embeddings of 5's from the handwritten digits data set. The figures show every 14th embedding of the sorting w.r.t. 100 digits for $\epsilon = 2.0$, and $\epsilon = 20.0$.

5 Missing Data

Failures of sensors, matching of databases with disjunct feature sets, or conditions where data can get lost (e.g., in outer space due to X-ray) are typical examples for practical scenarios, in which data sets are incomplete. Sensors may be out of order, or data may get lost. However, it might be desirable to compute a latent embedding of high-dimensional data. In this section we will introduce strategies that allow unsupervised nearest neighbor regression to cope with missing data. The question arises, if the embedding approach can exploit useful structural information to reconstruct the missing entries. Experimental analyses will answer these questions.

5.1 Imputation Methods

For the case that the distribution of missingness is conditionally independent of the missing values given the observed data, the data is denoted as *missing at random*[1]. Schafer and Graham [24] have reviewed methods to handle them. In case of scarce data sets joint densities can be computed in a probabilistic framework [9].

If possible, the method can directly deal with missing data (our embed-and-repair method that will be introduced in Section 5.3 belongs to this class). For SVM classification such an approach has been introduced by Chechik *et al.* [6] that alters the SVM margin interpretation to directly deal with incomplete patterns. But the method is best suited for features that are absent than those that are MNAR. An extension has been proposed by Dick *et al.* [7] who marginalize kernels over the assumed imputation distribution. The approach by Williams *et al.* [29] employs logistic regression for classification of incomplete data, and performs an analytic integration with an estimated conditional density function instead of imputation. The approach is interesting, as it does not only take into account the complete patterns, but also the incomplete patterns in a semi-supervised kind of way.

5.2 Repair-and-Embed

Let $\mathbf{Y}$ be the matrix of high-dimensional patterns. In the missing data scenario we assume that some patterns are incomplete, i.e., for one entry of $\mathbf{y}_j$ it holds $y_{ij} = n.a.$ We can treat the problem of missing entries as regression approach. First, we define $\tilde{\mathbf{Y}}$ as the matrix of complete patterns, i.e., it holds $\nexists y_{ij} = n.a.$ In contrast, $\mathbf{Y} \setminus \tilde{\mathbf{Y}}$ is the matrix of incomplete pattern. To complete $\mathbf{Y} \setminus \tilde{\mathbf{Y}}$ repair-and-embed trains a regression model $\tilde{\mathbf{f}}$ based on $\tilde{\mathbf{Y}}$. We propose to first fill the vectors $\mathbf{y}_j$ from $\mathbf{Y}$ with minimal number of missing values, and add the completed patterns to $\tilde{\mathbf{Y}}$ for repairing the next vectors with minimal number of missing entries in an iterative kind of way.

Let y_{ij} be the entry to complete. We can employ matrix $\tilde{\mathbf{Y}}_{-i}$ as training pattern[2], while $\tilde{\mathbf{y}}_i = y_{i1}, \ldots, y_{i\tilde{N}}$ comprises the corresponding labels. Entry y_{ij} is estimated

[1] Missing at random (MAR) means that entries are missing randomly with uniform distribution, in contrast to *missing not at random* (MNAR), where dependencies exist, e.g., the missingness depends on certain distributions.

[2] $\tilde{\mathbf{Y}}_{-i} = ((\mathbf{y}_1)_{-i}, \ldots, (\mathbf{y}_{\tilde{N}})_{-i})$ with $(\mathbf{y}_k)_{-i} = (y)_k$ with $k = 1, \ldots, d$ and $k \neq i$.

with $\tilde{\mathbf{f}}$ leading to the complete vector $\mathbf{y}_j$ that can be embedded as usual, depending on the employed approach (UNN, UNN$_g$, etc.). To summarize, repair-and-embed works as follows:

1. Choose $\mathbf{y}_j$ from $\mathbf{Y} \setminus \tilde{\mathbf{Y}}$ with minimal number of missing entries,
2. employ $\tilde{\mathbf{Y}}_{-i}$ as training pattern for prediction of y_{ij}, and predict with KNN regression ($\rightarrow \grave{\mathbf{y}}_j$),
3. add $\grave{\mathbf{y}}_j$ to $\tilde{\mathbf{Y}}$, and start from 1. until all patterns are complete,
4. embed all $\tilde{\mathbf{Y}}$ with UNN approach.

As KNN regression is a non-parametric method, no training is necessary, only K has to be chosen carefully.

5.3 Embed-and-Repair

The second variant for embedding incomplete data is to embed a vector $\mathbf{y}_j$ with missing entries at dimension i ignoring the i-th dimension during the embedding (the computation of the DSRE), i.e., minimizing:

$$E_{-i}(\mathbf{X}) = \frac{1}{N}\|\mathbf{Y}_{-i} - \mathbf{f}_{UNN}(\mathbf{x};\mathbf{X})_{-i}\|_F^2, \tag{7}$$

The approach starts iteratively with complete vectors $\mathbf{y}$ from $\tilde{\mathbf{Y}}$, and then incomplete patterns with increasing number of missing values. Starting the dimensionality reduction with complete patterns is reasonable to get as close as possible to the structure of the complete embedding. Embed-and-repair is a greedy approach that only considers the locally best embedding w.r.t. the available information. As the embedded pattern has to be completed to allow embeddings of further patterns, the gaps are closed with entries that ensure that the embedding is minimal w.r.t. to the overall DSRE. This is obviously the average of the K nearest points for dimension i, i.e., the nearest neighbor estimation $\mathbf{f}_{KNN}(x_i)$, see Equation 2.

Figure 9 illustrates the embed-and-repair strategy for neighborhood size $K = 2$. Pattern $\mathbf{y}^* = (y_1^*, \cdot)$ is incomplete. It is embedded at the position where it leads to the lowest DSRE w.r.t. the first dimension: between x', and x''. Then, the gap is filled with the mean of the second dimension of $\mathbf{y}'$, and $\mathbf{y}''$ yielding $(y_1^*, 0.5(y' + y''))$. To summarize, embed-and-repair works as follows:

1. Choose $\mathbf{y}_j$ from $\mathbf{Y}$ with minimal number of missing entries,
2. embed $\mathbf{y}_j$ with UNN/ UNN$_g$ minimizing $E_{-i}(\mathbf{X})$,
3. add $\mathbf{x}_j$ to $\tilde{\mathbf{X}}$,
4. complete $\mathbf{y}_j$ w.r.t. KNN based on $\tilde{\mathbf{X}}$ ($\rightarrow \grave{\mathbf{y}}_j$),
5. add $\grave{\mathbf{y}}_j$ to $\tilde{\mathbf{Y}}$,
6. start from 1 until $\mathbf{Y} \setminus \tilde{\mathbf{Y}}$ empty.

The difference of KNN imputation, and embed-and-repair imputation is that the embed-and-repair KNN prediction is based on neighborhoods in latent space. Hence, it is a dimensionality reduction-oriented imputation method based on characteristics introduced by UNN regression.

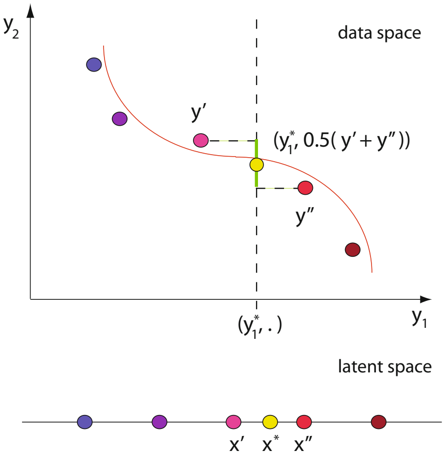


Fig. 9. Embed-and-repair: The incomplete pattern $\mathbf{y}^*$ is embedded at the position where it leads to the lowest DSRE w.r.t. the first dimension: between x', and x'', then the gap y_2^* is filled with KNN and $K = 2$

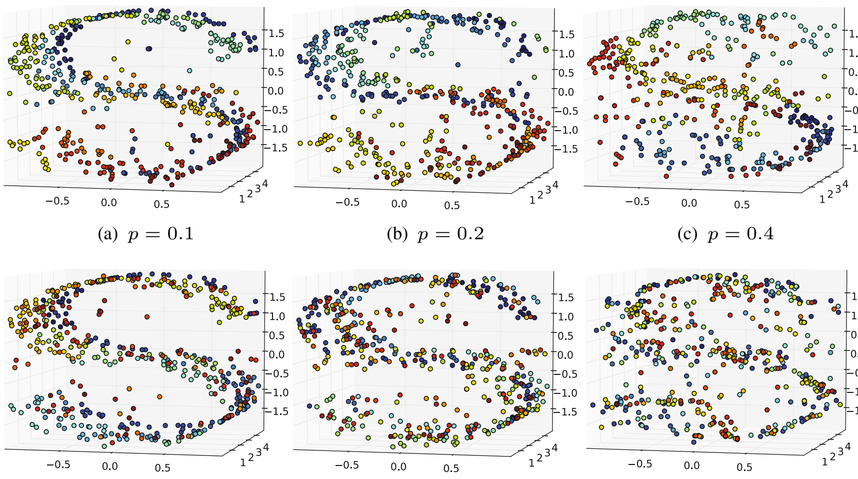
Table 4. Comparison of imputation Error E, and DSRE with UNN embedding with repair-and-embed (R-a-E) and embed-and-repair (E-a-R) on 3D-S, and 3D-S$_h$ w.r.t. increasing data missing rate p

		R-a-E		E-a-R		complete
data	p	E	DSRE	E	DSRE	UNN
3D-S	0.01	0.0507	**147.2**	**0.0269**	165.39	142.8
	0.1	0.3129	**143.8**	**0.2884**	265.2	142.8
	0.2	0.6454	**149.0**	**0.6146**	369.2	142.8
	0.3	0.9557	**152.7**	**0.9265**	452.3	142.8
3D-S$_h$	0.01	**0.0235**	**104.2**	0.0309	119.7	105.5
	0.1	0.2671	**101.9**	**0.2595**	217.6	105.5
	0.2	0.5509	**122.2**	**0.5007**	296.8	105.5
	0.3	0.8226	**129.5**	**0.5285**	301.8	105.5

5.4 Experimental Analysis

In the following, we describe the experimental setup for the comparison between both approaches. We generate the missing data scenario by removing entries y_{ij} from $\mathbf{Y}$ with probability p, and experimentally analyze the final DSRE in comparison to the DSRE for the complete reference matrix (without missing entries), and the imputation error $E = \sum_i^N \|\mathbf{y}_i^+ - \grave{\mathbf{y}}_i\|$, which is the deviation from the original complete patterns $\mathbf{y}_i^+$, and the repaired pendants $\grave{\mathbf{y}}$ of the incomplete versions.

Table 4 shows the experimental results for various degrees of missing data modeled by increasing p on the data set 3D-S. For embedding we employ UNN. The experimental results show that repair-and-embed achieves the lowest DSRE on both data sets. The results are even very close to the DSRE achieved on the data set without missing



(a) $p = 0.1$ (b) $p = 0.2$ (c) $p = 0.4$

(d) $p = 0.1$ (e) $p = 0.2$ (f) $p = 0.4$

Fig. 10. UNN embedding of 3D-S with missing data for missing rates $p = 0.1, 0.2$, and $p = 0.4$ for repair-and-embed (upper row), and for embed-and-repair (lower row)

values (complete). Embed-and-repair achieves the lowest imputation error E in seven of the eight cases, but much worse results for the DSRE. While the DSRE results are still satisfactory with 1% of incomplete data, the approach fails for higher missing rates. Obviously, it is difficult for MAR data to first determine the structure from incomplete patterns.

Figure 10 shows the embeddings of UNN on the 3D-S data set for increasing missing rates p, and repair-and-embed in the upper row. The patterns are colorized w.r.t. their latent colors (similar colors are neighbored). Hence, similar colors in data space show that the embedding has been successful. Figure 10 (a) shows the embedding with a low missing rate of $p = 0.1$. The figure shows that the 3D-S is almost completely reconstructed, while the colors of the embedding show that a reasonable learning process took place: neighbored solutions in latent space have similar colors. The higher the rate of incomplete data the worse is the embedding (different colors are neighbored). But higher missing rates can also be recognized by deviations from the S-structure.

The lower row shows the corresponding experimental results for UNN with embed-and-repair, also with increasing missing rates p. The colors show that the embeddings are worse for high missing rates. For example, Figure 10 (f) shows that the embedding for missing rate $p = 0.4$ with repair-and-embed is comparably bad, a result that is consistent with the DSRE of Table 4.

6 Conclusions

Fast dimensionality reduction methods are required that are able to process huge data sets, and large dimensions. With UNN regression we have fitted a well-known regression technique into the unsupervised setting for dimensionality reduction. The two iterative UNN strategies are efficient methods to embed high-dimensional data into fixed one-dimensional latent space taking linear time. UNN achieves lower DSREs, but UNN$_g$ is slightly faster because of the multiplicative constants of UNN. We concentrated on the employment of the ϵ-insensitive loss, and its influence on the DSRE. It could be observed that both iterative UNN regression strategies could benefit from the ϵ-insensitive loss, in particular the iterative variant UNN$_g$ could be improved employing a loss with $\epsilon > 0$. Obviously, local optima can be avoided.

We have introduced two algorithmic variants that allow the efficient embedding of incomplete data. The results have shown that the embeddings are better in case of repair-and-embed, as it is obviously difficult to determine the embedding of the data with incomplete patterns. From the perspective of imputation, first repairing incomplete data is a straightforward approach. In contrast, first embedding data at the location with lowest DSRE, and then repairing the entries employing the neighbors is an approach that makes use of the *intrinsic structure* UNN regression assumes for imputation leading to comparatively good pattern reconstructions, but worse embeddings than repair-and-embed.

References

1. Sdss 2011, sloan digital sky survey (2011), `http://www.sdss.org`
2. Baillard, A., Bertin, E., de Lapparent, V., Fouqué, P., Arnouts, S., Mellier, Y., Pelló, R., Leborgne, J.-F., Prugniel, P., Markarov, D., Makarova, L., McCracken, H.J., Bijaoui, A., Tasca, L.: Galaxy morphology without classification: Self-organizing maps, 532, A74, 1103.5734 (2011)
3. Bhatia, N., Vandana: Survey of nearest neighbor techniques. CoRR, abs/1007.0085 (2010)
4. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer (2007)
5. Carreira-Perpiñán, M.Á., Lu, Z.: Parametric dimensionality reduction by unsupervised regression. In: Computer Vision and Pattern Recognition (CVPR), pp. 1895–1902 (2010)
6. Chechik, G., Heitz, G., Elidan, G., Abbeel, P., Koller, D.: Max-margin classification of data with absent features. Journal of Machine Learning Research 9, 1–21 (2008)
7. Dick, U., Haider, P., Scheffer, T.: Learning from incomplete data with infinite imputations. In: International Conference on Machine Learning (ICML), pp. 232–239 (2008)
8. Fix, E., Hodges, J.: Discriminatory analysis, nonparametric discrimination: Consistency properties, vol. 4 (1951)
9. Ghahramani, Z., Jordan, M.I.: Supervised learning from incomplete data via an em approach. In: Advances in Neuronal Information Processing (NIPS), pp. 120–127 (1993)
10. Gieseke, F., Polsterer, K.L., Thom, A., Zinn, P., Bomanns, D., Dettmar, R.-J., Kramer, O., Vahrenhold, J.: Detecting quasars in large-scale astronomical surveys. In: ICMLA, pp. 352–357 (2010)
11. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer, Berlin (2009)
12. Hastie, Y., Stuetzle, W.: Principal curves. Journal of the American Statistical Association 85(406), 502–516 (1989)
13. Hull, J.: A database for handwritten text recognition research. IEEE PAMI 5(16), 550–554 (1994)
14. Jolliffe, I.: Principal component analysis. Springer series in statistics. Springer, New York (1986)

15. Kitchin, C.: Galaxies in Turmoil – The Active and Starburst Galaxies and the Black Holes That Drive Them. Springer, New York (2007)
16. Klanke, S., Ritter, H.: Variants of unsupervised kernel regression: General cost functions. Neurocomputing 70(7-9), 1289–1303 (2007)
17. Kramer, O.: Dimensionalty reduction by unsupervised nearest neighbor regression. In: Proceedings of the 10th International Conference on Machine Learning and Applications (ICMLA), pp. 275–278. IEEE Press (2011)
18. Kramer, O.: On unsupervised nearest-neighbor regression and robust loss functions. In: International Conference on Artificial Intelligence, pp. 164–170 (2012)
19. Lawrence, N.D.: Probabilistic non-linear principal component analysis with gaussian process latent variable models. Journal of Machine Learning Research 6, 1783–1816 (2005)
20. Meinicke, P.: Unsupervised Learning in a Generalized Regression Framework. PhD thesis, University of Bielefeld (2000)
21. Meinicke, P., Klanke, S., Memisevic, R., Ritter, H.: Principal surfaces from unsupervised kernel regression. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(9), 1379–1391 (2005)
22. Pearson, K.: On lines and planes of closest fit to systems of points in space. Philosophical Magazine 2(6), 559–572 (1901)
23. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326 (2000)
24. Schafer, J.L., Graham, J.W.: Missing data: Our view of the state of the art. Psychological Methods 7(2), 147–177 (2002)
25. Schölkopf, B., Smola, A., Müller, K.-R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10(5), 1299–1319 (1998)
26. Smola, A.J., Mika, S., Schölkopf, B., Williamson, R.C.: Regularized principal manifolds. Journal on Machine Learning Research 1, 179–209 (2001)
27. Tan, S., Mavrovouniotis, M.: Reducing data dimensionality through optimizing neural network inputs. AIChE Journal 41(6), 1471–1479 (1995)
28. Tenenbaum, J.B., Silva, V.D., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000)
29. Williams, D., Liao, X., Xue, Y., Carin, L., Krishnapuram, B.: On classification with incomplete data. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(3), 427–436 (2007)

Appendix

Test Problems

In this work we employ various test problems to evaluate UNN regression. The data sets are summarized in the following.

S-Structure

The three-dimensional S data sets consists of 500 points in the version 3D-S without hole. The counterpart with hole 3D-S$_h$ consists of approximately 350 points.

Digits

The *Digits* data set [13] comprises handwritten digits, and is often employed as reference problem related to the recognition of handwritten characters and digits. Figure 11 shows a collection of images from the *Digits* data set. The collection shows all ten digits, while the experimental analysis concentrates on a subset.

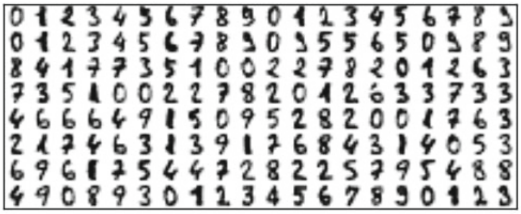


Fig. 11. Visualization of a collection of images from the *Digits* data set

Astronomy

Our experimental study is based on data from the Sloan Digital Sky Survey (SDSS) [1]. In [2] the EFIGI catalog of 4,458 nearby galaxies is presented. We concentrate on an EFIGI subset of 100 galaxies of types -6 to -4, and 7 to 11. The images are 40×40 RGB images. The data has been preprocessed: all entries below a threshold $\epsilon = 0.1$ have been set to 0.

Patient Classification and Automatic Configuration of an Intelligent Wheelchair

Brígida Mónica Faria[1,2], Luís Paulo Reis[3,4], Nuno Lau[1], João Couto Soares[4],
and Sérgio Vasconcelos[4]

[1] Dep. Elect., Telecomunicações e Informática (DETI/UA), Inst. Eng. Electrónica e Telemática de Aveiro (IEETA), Universidade de Aveiro, Aveiro, Portugal
{monica.faria,nunolau}@ua.pt
[2] Escola Superior de Tecnologia da Saúde do Porto, Instituto Politécnico do Porto (ESTSP/IPP), Vila Nova de Gaia, Portugal
[3] Escola de Engenharia da Universidade do Minho - DSI, Universidade do Minho (EEUM/DSI), Guimarães, Portugal
lpreis@dsi.uminho.pt
[4] Laboratório de Inteligência Artificial e Ciência de Computadores (LIACC), Universidade do Porto, Porto, Portugal
{joao.couto.soares,ei05074}@fe.up.pt

Abstract. The access to instruments that allow higher autonomy to people is increasing and the scientific community is giving special attention on designing and developing such systems. Intelligent Wheelchairs (IW) are an example of how the knowledge on robotics and artificial intelligence may be applied to this field. IWs can have different interfaces and multimodal interfaces enabling several inputs such as head movements, joystick, facial expressions and voice commands. This paper describes the foundations for creating a simple procedure for extracting user profiles, which can be used to adequately select the best IW command mode for each user. The methodology consists on an interactive wizard composed by a flexible set of simple tasks presented to the user, and a method for extracting and analyzing the user's execution of those tasks. The results showed that it is possible to extract simple user profiles, using the proposed method.

Keywords: Classification, Patient, Intelligent Wheelchair, Knowledge Discovery.

1 Introduction

The population with physical disabilities has earned more relevance and has attracted the attention of international health care organizations, universities and companies interested in developing and adapting new products. The current tendency reflects the demand for an increase on health and rehabilitation services, in a way that senior and handicapped individuals might become more and more independent when performing quotidian tasks.

Regardless the age, mobility is a fundamental characteristic for every human being. Children with disabilities are very often deprived of important opportunities and face

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 268–282, 2013.

serious disadvantages compared to other children. Adults who lose their independent means of locomotion become less self-sufficient, raising a negative attitude towards themselves. The loss of mobility originates obstacles that reduce the personal and vocational objectives [1]. Therefore it is necessary to develop technologies that can aid this population group, in a way to assure the comfort and independence of the elderly and handicapped people. Wheelchairs are important locomotion devices for those individuals. There is a growing demand for safer and more comfortable wheelchairs, and therefore, a new Intelligent Wheelchair (IW) concept was introduced. However, most of the Intelligent Wheelchairs developed by distinct research laboratories [1] have hardware and software architectures very specific for the used wheelchair model/developed project and are typically very difficult to configure in order for the user to start using them.

The paper is organized as follows: Section 2 presents the state of art on intelligent wheelchairs and the IntellWheels project. Section 3 contains a description of the methodology for automatically extracting the users' profiles in order to give the best interface. The implementation of the system is presented in section 4 and the experiments and results achieved are presented in section 5. Finally some conclusions and future work are described in the last section.

2 Intelligent Wheelchairs

In the last years several prototypes of Intelligent Wheelchairs have been developed and many scientific work has been published [2] [3] in this area. Simpson [1] provides a comprehensive review of IW projects with several descriptions of intelligent wheelchairs. The main characteristics of an IW are [2] [4]: autonomous navigation with safety, flexibility and obstacle avoidance capabilities; communication with others devices such automatic doors and other wheelchairs and interaction with the user using distinct types of devices such as joysticks, voice interaction, vision and other sensor based controls like pressure sensors.

2.1 Intelligent Wheelchairs' Projects

The first project of an autonomous wheelchair for physical handicapped was proposed by Madarasz in 1986 [5]. It was planned as a wheelchair with a micro-computer, a digital camera and an ultra-sound scanner with the objective of developing a vehicle that could move around in populated environments without human intervention. Hoyer and Holper [6] presented a modular control architecture for an omni-directional wheelchair. The characteristics of NavChair [7], such as the capacity of following walls and avoid obstacles by deviation are described in [7-9]. Miller and Slak [10] [11] proposed the system Tin Man I with three operation modes: one individual driving a wheelchair with automatic obstacles deviation; moving through-out a track and moving to a point (x,y). This kind of chair evolved to Tin Man II which included advanced characteristics such as storing travel information, return to the starting point, follow walls, pass through doors and recharge battery. Wellman [12] proposed a hybrid wheelchair equipped with two extra legs in addition to its four wheels, to allow stair climbing and movement on rough terrain. FRIEND is a robot for rehabilitation which consists of a motorized wheelchair and a MANUS manipulator [13] [14]. In this case, both the vehicle and the manipulator are controlled by voice commands. Some projects present solutions for quadriplegic individuals, where facial expressions recognition is used to control the wheelchair [4] [15] [16]. In 2002, Pruski presented VAHM, a user adapted intelligent wheelchair [17].

Satoh and Sakaue [18] presented an omni-directional stereo vision-based IW which detects both the potential hazards in a moving environment and the postures and gestures of a user, using a stereo omni-directional system, which is capable of acquiring omni-directional color image sequences and range data simultaneously in real time. In 2008 John Spletzer studied the performance of LIDAR based localization for docking an IW system [19] and in 2009 Horn and Kreutner [20] showed how the odometric, ultrasound, and vision sensors are used in a complementary way in order to locate the wheelchair in a known environment. In fact, the research on IW has suffered a lot of developments in the last few years. Some IW prototypes may be even controlled using "thoughts". This type of technology uses sensors that pick up electromagnetic waves of the brain [21-23] and are able to detect patterns on the brain waves that may be associated with the users' desired commands.

2.2 IntellWheels Project

This section presents a brief overview of the Intelligent Wheelchair project that is being developed at the University of Porto in collaboration with INESC-P, University of Aveiro, University of Minho, IPP and APPC in Portugal. The main objective of the IntellWheels Project is to develop an intelligent wheelchair platform that may be easily adapted to any commercial wheelchair and aid any person with special mobility needs. Initially, an evaluation of distinct motorized commercial wheelchair platforms was carried out and a first prototype was developed in order to test the concept. The first prototype was focused on the development of the modules that provide the interface with the motorized wheelchair electronics using a portable computer and other sensors. Several different modules have been developed in order to allow different ways of conveying commands to the IW. These include, for example, joystick control with USB, voice commands, control with head movements and gestures, and facial expressions recognition. Fig. 1 shows the input devices already available in the IntellWheels IW.

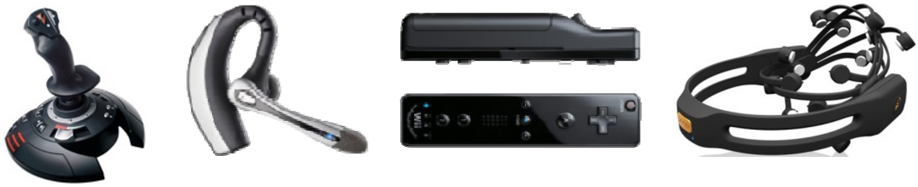


Fig. 1. Joystick, headset with microphone, Nintendo Wii Remote and Emotiv Brain Computer Interface [24]

The project research team considered the difficulty that some patients have while controlling a wheelchair using traditional input devices such as the traditional joystick. Therefore, new ways of interaction between the wheelchair and the user have been integrated, creating a system of multiple entries based on a multimodal interface. The system allows users to choose which type of command best fits their needs, increasing the level of comfort and safety.

A simulated environment was developed that models the intelligent wheelchair and its environment. In this environment it is possible to test in a safe manner the different ways of driving the intelligent wheelchair, since the behavior of the simulated intelligent wheelchair is very identical to the behavior of the real intelligent wheelchair.

3 Methodology for Automatic Extraction of User Interfaces/Profiles

The potential users of the Intelligent Wheelchair have particular characteristics and constraints. Therefore it is very important to adjust and adapt the way of driving the intelligent wheelchair to the specific patient. The data acquired when the users are performing a test drive using a multimodal interface and an intelligent wheelchair will allow improving the adaptability.

This section presents the features and the global architecture developed of the IntellWheels Multimodal Interface.

3.1 IntellWheels Multimodal Interface

There are several publications in the literature of projects related to the issue of adapting and designing specific interfaces for individuals with severe physical disabilities [25-27]. Nevertheless, most of these projects present restricted solutions concerning the accessibility to the user to drive a particular wheelchair. It is common to find just one solution such as voice recognition, while other focus merely on facial expressions recognition [27]. Since the physical disability is very wide and specific to each individual, it becomes important to provide the greatest possible number of recognition methods to try to cover the largest possible number of individuals with different characteristics.

The IntellWheels Multimodal Interface offers five basic input devices: joystick, speech recognition, recognition of head movements and gestures, the use of a generic gamepad and facial expressions. In addition, IntellWheels project proposes an architecture that makes the interface extensible enabling the addition of new devices and recognition methods in an easy way. It also presents a flexible paradigm that allows the user to define the sequences of inputs to assign to each action, allowing for an easy and optimized configuration for each user. For example an action of following the right wall can be triggered by blinking the left eye followed by the expression "go".

Fig. 2 shows the IntellWheels Multimodal Interface where all the input devices are connected.

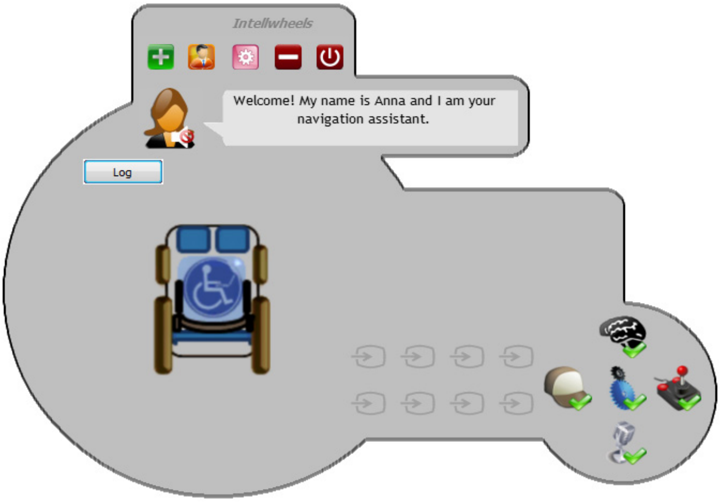

Intellwheels
Welcome! My name is Anna and I am your navigation assistant.
Log

Fig. 2. IntellWheels Multimodal Interface

3.2 Multimodal Data Gathering System

Based on the IntellWheels prototype and using the real and simulated environments, this work is focused on devising appropriate data gathering and data analysis systems that enable the construction of patient and environment models using knowledge discovery methodologies. The constructed models will be used together with a simple interface library in the context of an interface selector application that will also use knowledge discovery methodologies in order to select and configure the most appropriate multimodal interface for each patient in each specific situation.

The multimodal data gathering system enables the collection of real-time input information from patients with distinct disabilities. The system also enables the collection of environmental information and more high-level information concerning the wheelchair localization and orientation, task in execution, among other information.

Considering the concept of flexibility and multimodality of the IntellWheels project, the required data to collect from the platform may come from many sources: input devices, sensors (both real and virtual) and the Simulator. The Fig. 3 presents the software architecture.

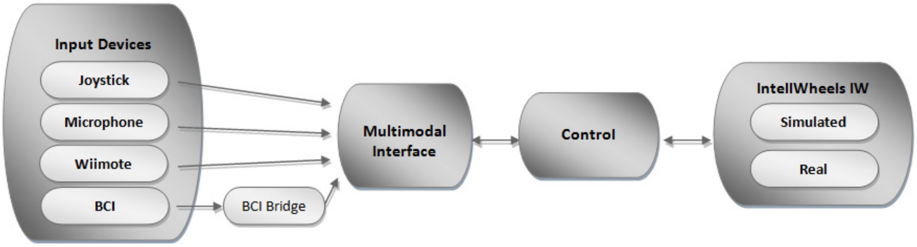


Fig. 3. IntellWheels' software architecture

The control application can connect to both the real IW or the Simulator, gathering and processing data from their sensors. The control works as the server side regarding data communication with the Multimodal Interface. The Multimodal Interface, in turn, acts as the server side concerning the input devices connections, since the Multimodal Interface manages all the input devices.

The data acquisition system is distributed among the Control application, the Multimodal Interface and the input devices bridge applications. As such, one file with captured data is created by each application.

Data Synchronization. In order to synchronize the files, a timestamp is attached to all information acquired. The information required for the synchronization concerns the IntellWheels platform uptime. For this reason and since the applications are not executed at the same time, a flow to set the same uptime for all applications was created: the Control application, the first one to be executed, sends its uptime to the Multimodal Interface, which in turn sends this value to all input devices' bridge applications. Each application has a time delta variable which stores the difference between its own uptime and the Control's uptime. The time delta variable is updated several times throughout the acquisition process. After a certain amount of inputs is received from the Multimodal Interface by the Control application, it again sends a message with its current uptime, which once more is distributed to all applications by the flow previously explained.

Data File Format. To save the data, an extensible markup language (XML) type file format was chosen to be used because of its flexibility.

The header of the file contains the description of each type of data the application gathers. An example of an input data collected file can be seen next:

```
<MMI_LOG>
 <INPUTS>
  <item>
   <id> wiimote </id>
   <label> VelX; VelY; VelZ; Button N;Battery Level %;Error
</label>
  </item>
 </INPUTS>
 <DATA>
  <item>
   <timestamp> 579.6243 </timestamp>
   <input> wiimote </input>
   <values> -10; -15; 0; ; 78; </values>
  </item>
 </DATA>
</MMI_LOG>
```

3.3 Data Gathering Process

The sample of individuals includes patients with distinct disabilities (Thrombosis, Stroke, Cerebral Palsy, Parkinson, Alzheimer and Multiple Sclerosis, among others).

The data collecting process was divided into two parts. In each of the parts, the patients are asked to perform distinct inputs:

- Perform obligatory inputs, including a complete set of a previously specified protocol: voice commands; facial expressions; head, arm and hand movements. The objective of this protocol is to make a profile of the users.
- Perform free inputs, which enable the patient to perform some given tasks but using its own and completely free preferred process.

The inputs are performed in distinct environmental conditions: noise and lighting variations; distinct pavement and wheelchair movements; tasks performed in parallel (such as maintaining a conversation). Tracing a user diagnostic can be very useful to adjust certain settings allowing for an optimized configuration and improved interaction between the user and the multimodal interface.

Accordingly, the Intellwheels Multimodal Interface contains a module capable of performing series of training sessions, composed of small tests for each input modality. These tests may consist, for example, of asking the user to press a certain sequence of buttons on the gamepad, or to move one of the gamepads' joysticks to a certain position. Another test may consist in asking the user to pronounce a set of voice commands, or to perform a specific head movement.

The tests should be performed sequentially and should have an increasing difficulty. Additionally, the tests should be reconfigurable and extensible. Finally, the tests sets and theirs results should be saved on a database, accessible by the Intellwheels Multimodal Interface. Therefore, the following user characteristics should be extracted and these characteristics can be separated in two different types: quantitative and qualitative. The quantitative measures consist of: the time taken to perform a full button sequence; the average time between pressing two buttons; the average time to place a gamepad analogical switch on a certain position; the average time to position the head on a certain position; the trust level of speech recognition; maximum amplitude achieved with the gamepad analogical switches in different directions; maximum amplitude achieved with the head in different directions and number of errors made using the gamepad. Using the quantitative measures, the following qualitative measures should be estimated: user ability to use the gamepad buttons; user ability to perform head movements and user ability to pronounce voice commands.

At the end of the training session, the tracked user information should be saved to an external database, containing the users' profile. The user profile can be used to improve security, by defining, for each user, a global trust level for each input modality. The trust level can be used to advice the user of which modality to use, at the creation of a new association. Also, it could be useful to activate confirmation events whenever a user requests a certain output action using an input level with a low trust level.

Another functionality of the user´s profile is capturing the EEG signals using a brain computer interface [27] for using as input facial expressions and thoughts.

The users will be asked to fill a complete questionnaire about the experiment and their preferences regarding each control method for each task in each environmental condition.

Also, a simple library of wheelchair interfaces is being developed together with an application that enables fast generation and configuration of these interfaces. The interface selection application will be based on the use of machine learning algorithms that will use the available patient and environment models to select the most appropriate interfaces from the interface library.

4 Implementation

This section presents the implementation for the proposed User Profile feature. Firstly, it explains the approach followed to specify which test sets are going to be loaded by the module responsible for tracking the users' profile. Secondly, we show the simple profiling methods that were implemented to create the future user classification. Next, we will present how the extracted information was used to adjust certain settings of the interface. Finally, a demonstration of how the profile is stored to enable future use is also made.

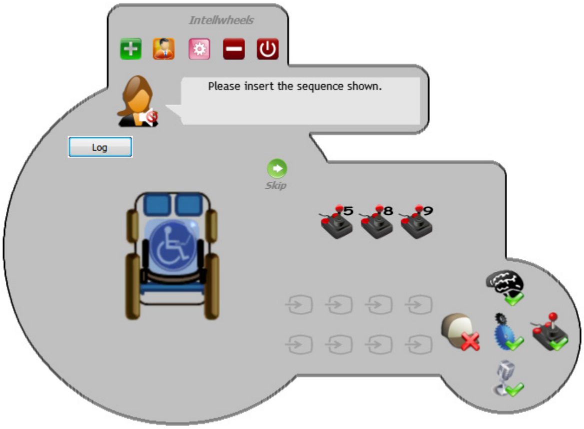
4.1 Definition of the Sets

To perform the measures previously described, a simple XML grammar was defined. It implements four configurable distinct test types: sequences of gamepad buttons; voice commands; positions for both joysticks and positions for head.

Example of XML containing user profile test set:

```
<INTELLWHEELS_PROFILER>
 <BINARY_JOYSTICK>
  <item>
   <sequence> joystick.1
      joystick.2 </sequence>
   <difficulty> easy </difficulty>
  </item>
 </BINARY_JOYSTICK>
 <ANALOG_JOYSTICK>
 (…)
 <ANALOG_WIIMOTE>
  <item> <x>100</x> <y>0</y></item>
 </ANALOG_WIIMOTE>
 <SPEECH>
  <item> go forward </item>
  <item> turn right </item>
  <item> create new sequen </item>
  <item> stop </item>
 </SPEECH>
</INTELLWHEELS_PROFILER>
```

The proposed XML grammar makes it possible for an external operator to configure the test set that they find most appropriate for a specific context or user. When a user starts the training session, the four different types of tests are iterated. In order to attain a consistent classification of the user, the defined grammar should be sufficiently extensive. The test set specified on the XML file is iteratively presented to the user. It starts by asking the user to perform the gamepad button sequence as can be observed in Fig. 4.

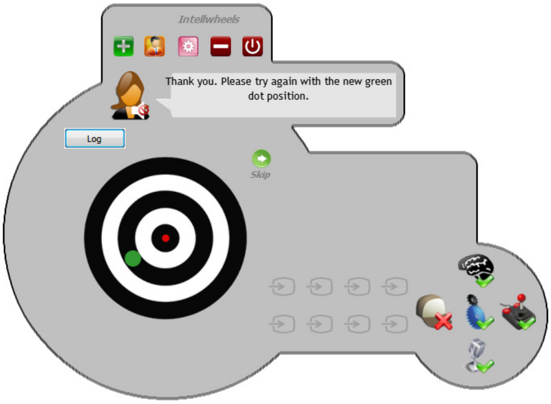


Please insert the sequence shown.

Fig. 4. User profiler gamepad and voice tests

When the user ends the first component of the user profiler module, the navigation assistant asks the user to pronounce the voice commands stored in the XML. Also, the quantitative results for the gamepad buttons test are presented.

The last part of the user profiler test is shown in Fig. 5. The user is invited to place the gamepad's joystick into certain positions. A similar approach is used for the head movements test.



Thank you. Please try again with the new green dot position.

Fig. 5. User profiler joystick test

To define the user proficiency in using the gamepad buttons, a simple method was implemented. Each sequence defined on the grammar should have an associated difficulty level (easy, medium or hard). The difficulty type of a sequence may be related to its size, and to the physical distance between the buttons on the gamepad. Since the layout of a generic gamepad may change depending on the model, defining whether or not a sequence is of easy, medium or difficulty level is left to the operator.

When the user completes the gamepad sequences training part, an error rate is calculated for each of the difficulty levels. If these rates are higher than a minimum acceptable configurable value, the user classification in this item is immediately defined. This classification is then used to turn on the security feature, which is characterized by a confirmation event performed by the navigation assistant. For a grammar with 5 sequences of difficulty type easy, the maximum number of accepted errors would be 1. If the user fails more than one sequence, the confirmation event is triggered for any input sequence, of any difficulty type, and the gamepad training session is terminated. If the error rate for the easy type is less than 20% (=1/5) the training with the sub-set composed by the sequences of medium difficulty is initiated. At the end, a similar method is applied. If the error rate for the medium level is higher than 30%, the confirmation is triggered for the medium and hard levels of difficulty, and the training session is terminated. Finally, if the user makes it to the last level of difficulty, the training for the hard sequences sub-set is started. If the error rate is higher than 50%, the confirmation event is triggered only for sequences with a hard difficulty level. The best scenario takes place when the user is able to surpass the maximum accepted error rates for all the difficulty levels. In this situation, the confirmation event is turned off, and an output request is immediately triggered for any kind of input sequence composed only by gamepad buttons.

Defining the ideal maximum acceptable error rates is not easy. With this in mind, we made it possible to also configure these values in the XML grammar.

The joystick phase of the training session can be used to calculate the maximum amplitude achieved by the user. This value can then be used to parameterize the maximum speed value. Imagining a user who can only push the joystick to 50% of its maximum amplitude, the speed can be calculated by multiplying the axis value by two. This feature was not implemented. However, all the background preparation to implement it was set for future work.

The speech component of the training session was used to define the recognition trust level for each of the voice commands. The trust level is a percentage value retrieved by the speech recognition engine. This value is used to set the minimum recognition level for the recognition module.

Finally, the head movement phase of the training session has a similar purpose to the joystick's phase. Additionally, the maximum amplitude for each direction can be used to determine the range that will trigger each one of the leaning inputs of the head gestures recognition.

An extension of this profiling is related to the facial expressions and thoughts. A brain computer interface (BCI) was incorporated which can recognize the facial expressions and thoughts. However several patients suffering of cerebral palsy for example, are not able to produce all the facial expressions. For that reason it is also implemented a component in the profiling for testing the facial expressions (and even the thoughts) and where all the brain activity is recorded using the 14 sensors in the BCI for posterior analysis.

5 Experiments and Results

The main objective of the experimental work was to make a preliminary study of the tasks that can be implemented and the responses of the individuals in order of get information for the user profiling. The experiments involved 33 voluntaries, with a mean age of 24, a standard deviation of 4.2 and without any movements' restrictions.

The first experiment consisted in performing the sequence tasks with several levels of difficulty. In the first sequence the users needed to push the gamepad buttons GP1 - GP2 (easy level of difficulty); the second sequence was GP3 - GP8 (easy level of difficulty); the third sequence was GP5 - GP8 - GP9 (medium level of difficulty) and the last sequence was GP6 - GP1 - GP7 - GP4 - GP2 (hard level of difficulty). For the experiments with voice commands the individuals had to pronounce the sentences: "Go forward"; "Go back"; "Turn right"; "Turn left"; "Right spin"; Left Spin" and "Stop" to get the information about the recognition trust level for each voice command.

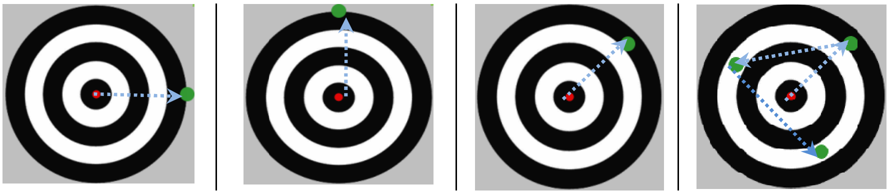


Fig. 6. User profiler joystick tests

The last two experiments involved the precision of the gamepad's joystick and the head movements. The voluntaries should move the small dot into the bigger one with the gamepad's joystick and with the wiimote controller. Fig. 6 shows some of the tasks that were asked. The positions were moving right; up; down; northeast; northwest; southeast and a sequence northeast - northwest - southeast without going back to the initial position in the center of the target.

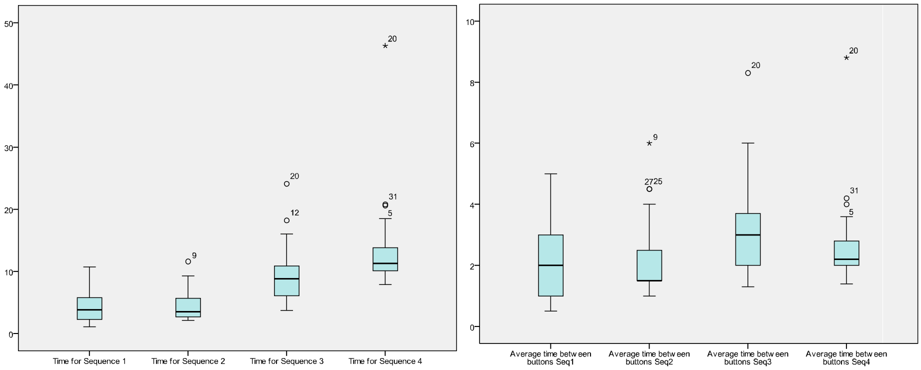


Fig. 7. Time for performing the sequences and average time between gamepad buttons

In general, the achieved results show the good performance of the individuals using gamepad and voice commands. The behaviour with head movements reflects more asymmetry and heterogeneous results, since several moderate and severe outliers exist in the time results. The time consumed to perform the sequences confirmed the complexity of the tasks as can be seen in Fig. 7. In terms of average time between buttons (Fig. 7) it is interesting to notice the results for the last sequence. Although it is more complex and longer it has a positive asymmetry distribution. This probably reveals that training may improve the user's performance.

In terms of errors, the third sequence presents a higher result with at least one fail. The last sequence presented a case where 12 errors were committed.

Table 1. Contingency table with the errors of sequences using gamepad

	Number of Errors							
Seq	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**12**
1	30	1	2	0	0	0	0	0
2	31	2	0	0	0	0	0	0
3	20	7	3	1	1	0	1	0
4	27	1	1	1	0	2	0	1

Table 2 presents several descriptive statistics, such as central tendency (mean, median) and dispersion (standard deviation, minimum and maximum), for the trust level of speech recognition.

Table 2. Descriptive Statistics for the trust level of speech recognition

Sentence	Mean	Median	S. Dev	Min	Max
"Go Forward"	**95.36**	**95.50**	0.51	93.9	95.9
"Go Back"	94.37	95.00	2.44	82.2	95.9
"Turn Right"	95.31	95.40	**0.42**	94.4	95.9
"Turn Left"	94.76	95.20	1.42	88.4	95.8
"Left Spin"	93.69	94.90	2.88	83.1	95.8
"Right Spin"	94.82	95.00	1.25	89.7	97.2
"Stop"	**92.67**	94.30	**3.85**	82.2	95.8
Total Sentences	94.43	94.99	1.08	92.24	95.93

The speech recognition has very good results. In fact, the minimum of minimums was 82.2 for the sentences "Go Back" and "Stop". The expression "Go Forward" has the highest mean and median. The sentence "Stop" is more heterogeneous since it has the higher standard deviation (3.85).

The paired samples t test was applied with a significance level of 0.05 to compare the means of time using joystick and head movements. The null hypothesis was established: the means of time to perform the target tasks with joystick and head movements were equal. The alternative hypothesis is: the means of time to perform the target tasks with joystick and head movements were different. The achieved power was of 0.80 with an effect size of 0.5. Table 3 contains the p values of the paired sample t tests and the 95% confidence interval of the difference. Observing the results for the positions Down and Northwest, it is valid to claim there are statistical evidences to affirm that the mean of time with joystick and head movements is different. This reveals the different performance by using in the same experience the joystick and the head movements.

Table 3. Confidence intervals of the difference and p values

	95% Confidence Interval of the difference		
Move the red dot to:	**Lower**	**Upper**	**P value**
Right	-2.29	0.67	0.273
Up	-1.38	0.08	0.080
Down	-9.67	-1.87	0.005*
Northeast	-2.89	0.66	0.211
Northwest	-2.74	-0.17	0.028*
Southeast	-6.26	1.00	0.150
Northeast - Northwest - Southeast	-5.32	0.37	0.085

Clustering analysis is a technique that can be used to obtain the information about similar groups. In the future, this can be used to extract characteristics for classification and users' profiling.

The results obtained by hierarchical clustering, using the nearest neighbour method and squared Euclidean distance, show the similar performance of subjects except one individual. In this case, using the R-square criteria, the number of necessary clusters to achieve 80% of the total variability retain by the clusters is 12. Since the sample of volunteers was from the same population, this kind of conclusions are very natural. So the next step will consist in obtain information about handicapped people. In fact, if the clusters of subjects could be defined then it should be interesting to work with supervised classification in which the best command mode would be the class.

6 Conclusions and Future Work

Although many Intelligent Wheelchair prototypes are being developed in several research projects around the world, the adaptation of user interfaces to each specific patient is an often neglected research topic. Typically, the interfaces are very rigid and adapted to a single user or user group. The Intellwheels project is aiming at developing a new concept of Intelligent Wheelchair controlled using high-level commands processed by a multimodal interface. However, in order to fully control the wheelchair, users must have a wheelchair interface adapted to their characteristics. In order to collect the characteristics of individuals it is important to have variables that can produce a user profile. The first stage must be a statistical analysis to extract knowledge of user and the surrounding.

The second stage must be a supervised classification to use Machine Learning algorithms in order to construct a model for automatic classification of new cases.

This paper mainly refers to the proposal of a set of tasks for extracting the required information for generating user profiles. A preliminary study has been done with several voluntaries, enabling to test the proposed methodology before going to the field and acquiring information with disabled individuals. In fact, this will be the next step of future work. The test set presented in this paper will be tested by a group of disabled individuals, and the results of both experiments will be compared to check if the performances of both populations are similar. Also, in order to collect feedback regarding the system usability, disabled users will be invited to drive the wheelchair in a number of real and simulated scenarios.

Acknowledgements. The authors would like to acknowledge to FCT – Portuguese Science and Technology Foundation for the INTELLWHEELS project funding (RIPD/ADA/109636/2009), for the PhD Scholarship FCT/SFRH/BD/44541/2008, LIACC – Laboratório de Inteligência Artificial e de Computadores, DETI/UA – Dep. Electrónica, Telecomunicações e Informática, IEETA – Instituto de Engenharia Electrónica e Telemática de Aveiro and ESTSP/IPP – Escola Superior de Tecnologia da Saúde Porto – IPP.

References

1. Simpson, R.C.: Smart wheelchairs: A literature review. Journal of Rehabilitation Research and Development 42(4), 423–436 (2005)
2. Braga, R.A.M., Petry, M., Moreira, A.P., Reis, L.P.: Concept and Design of the Intellwheels Platform for Developing Intelligent Wheelchairs. In: Cetto, J.A., Ferrier, J.-L., Filipe, J. (eds.) Informatics in Control, Automation and Robotics. LNEE, vol. 37, pp. 191–203. Springer, Heidelberg (2009)
3. Reis, L.P., Braga, R.A.M., Sousa, M., Moreira, A.P.: IntellWheels MMI: A Flexible Interface for an Intelligent Wheelchair. In: Baltes, J., Lagoudakis, M.G., Naruse, T., Ghidary, S.S. (eds.) RoboCup 2009. LNCS, vol. 5949, pp. 296–307. Springer, Heidelberg (2010)
4. Jia, P., Hu, H., Lu, T., Yuan, K.: Head Gesture Recognition for Hands-free Control of an Intelligent Wheelchair. Journal of Industrial Robot 34(1), 60–68 (2007)
5. Madarasz, R.L., Heiny, L.C., Cromp, R.F., Mazur, N.M.: The design of an autonomous vehicle for the disabled. IEEE Journal of Robotics and Automation 2(3), 117–126 (1986)
6. Hoyer, H., Hölper, R.: Open control architecture for an intelligent omnidirectional wheelchair. In: Proc.1st TIDE Congress, Brussels, pp. 93–97 (1993)
7. Levine, S.P., Bell, D.A., Jaros, L.A., Simpson, R.C., Koren, Y.: The NavChair assistive wheelchair navigation system. IEEE Transactions on Rehabilitation Engineering 7, 443–451 (1999)
8. Simpson, R.C., Levine, S.P., Bell, D.A., Jaros, L.A., Koren, Y., Borenstein, J.: NavChair: An Assistive Wheelchair Navigation System with Automatic Adaptation. In: Mittal, V.O., Yanco, H.A., Aronis, J., Simpson, R.C. (eds.) Assistive Technology and Artificial Intelligence. LNCS (LNAI), vol. 1458, pp. 235–255. Springer, Heidelberg (1998)
9. Bell, D.A., Borenstein, J., Levine, S.P., Koren, Y., Jaros, J.: An assistive navigation system for wheelchairs based upon mobile robot obstacle avoidance. In: IEEE Conf. on Robotics and Automation, pp. 2018–2022 (1994)

10. Miller, D., Slack, M.: Design and testing of a low-cost robotic wheelchair. Autonomous Robots 2, 77–88 (1995)
11. Miller, D.P.: Assistive Robotics: An Overview. In: Mittal, V.O., Yanco, H.A., Aronis, J., Simpson, R.C. (eds.) Assistive Technology and AI. LNCS (LNAI), vol. 1458, pp. 126–136. Springer, Heidelberg (1998)
12. Wellman, P., Krovi, V., Kumar, V.: An adaptive mobility system for the disabled. In: Proc. IEEE Int. Conf. on Robotics and Automation (1994)
13. Borgerding, B., Ivlev, O., Martens, C., Ruchel, N., Gräser, A.: FRIEND: Functional robot arm with user friendly interface for disabled people. In: 5th European Conf. for the Advancement of Assistive Technology (1999)
14. Volosyak, I., Ivlev, O., Graser, A.: Rehabilitation robot FRIEND II - the general concept and current implementation. In: ICORR 2005 - 9th International Conference on Rehabilitation Robotics, Chicago, pp. 540–544 (2005)
15. Ng, P.C., De Silva, L.C.: Head gestures recognition. In: Proceedings International Conference on Image Processing, pp. 266–269 (2001)
16. Adachi, Y., Kuno, Y., Shimada, N., Shirai, N.: Intelligent wheelchair using visual information on human faces. In: International Conference in Intelligent Robots and Systems, vol. 1, pp. 354–359 (1998)
17. Pruski, A., Ennaji, M., Morere, Y.: VAHM: A user adapted intelligent wheelchair. In: Proceedings of the 2002 IEEE International Conference on Control Applications, Glasgow, pp. 784–789 (2002)
18. Satoh, Y., Sakaue, K.: An Omnidirectional Stereo Vision-Based Smart Wheelchair. EURASIP Journal on Image and Video, 11 (2007)
19. Gao, C., Hoffman, I., Miller, T., Panzarella, T., Spletzer, J.: Performance Characterization of LIDAR Based Localization for Docking a Smart Wheelchair System. In: International Conference on Intelligent Robots and Systems, San Diego (2008)
20. Horn, O., Kreutner, M.: Smart wheelchair perception using odometry, ultrasound sensors and camera. Robotica 27(2), 303–310 (2009)
21. Hamagami, T., Hirata, H.: Development of Intelligent Wheelchair acquiring autonomous, cooperative and collaborative behaviour. In: IEEE International Conference on Systems Man and Cybernetics, vol. 4, pp. 3525–3530 (2004)
22. Lakany, H.: Steering a wheelchair by thought. In: IEE Digest, The IEE International Workshop on Intelligent Environments, vol. 2005(11059), pp. 199–202 (2005)
23. Rebsamen, B., Rebsamen, B., Burdet, E., Guan, C., Zhang, H., Teo, C.L., Zeng, Q., Ang, M., Laugier, C.: A Brain-Controlled Wheelchair Based on P300 and Path Guidance. In: IEEE/RAS-EMBS Int. Conference, vol. 20, pp. 1101–1106 (2006)
24. Emotiv, Emotiv Software Development Kit - User Manual for Release 1.0.0.4 (2011)
25. Philips, J., Millan, J., Vanacker, G., Lew, E., Galán, F., Ferrez, P., Van Brussel, H., Nuttin, M.: Adaptive shared control of a brain-actuated simulated wheelchair. In: 10th IEEE International Conference on Rehabilitation Robotics, Noordwijk (2007)
26. Matsumoto, Y., Ino, T., Ogasawara, T.: Development of Intelligent Wheelchair System with Face and Gaze Based Interface. In: 10th IEEE International Workshop on Robot and Human Communication, Bordeaux (2001)
27. Ju, S., Shin, Y., Kim, Y.: Intelligent wheelchair (iw) interface using face and mouth recognition. In: 13th International Conference on Intelligent User Interfaces, Canary Islands (2009)

An Approach on Merging Agents' Trust Distributions in a Possibilitic Domain

Sina Honari[1], Brigitte Jaumard[1], and Jamal Bentahar[2]

[1] Department of Computer Science and Software Engineering, Concordia University,
1455 De Maisonneuve Blvd. West, Montreal, Canada
[2] Concordia Institute for Information System Engineering, Concordia University,
1515 Ste-Catherine Street West, Montreal, Canada
s_hona@encs.concordia.ca, bjaumard@cse.concordia.ca,
bentahar@ciise.concordia.ca

Abstract. In this paper, we propose a novel approach on merging the trust distributions of an explorer agent in its advisors with the trust distributions of the advisor agents in a target agent. These two sets of merged distributions represent the trust of different, yet connected, agents in a multi-agent system. The deduced distribution measures an approximation of the explorer agent's trust in the target agent. The proposed approach can serve as a building block for estimating the trust distribution of an agent of interest in the multi-agent systems, who is accessible indirectly through a set of sequentially connected agents.

A common issue of modelling trust is the presence of uncertainty, which arises in scenarios where there is either lack of adequate information or variability in an agent's level of trustworthiness. In order to represent uncertainty, possibility distributions are used to model trust of the agents.

Keywords: Possibility theory, Uncertainty, Trust, Multi-agent systems.

1 Introduction

Social networking sites have become the preferred venue for social interactions. Despite the fact that social networks are ubiquitous on the Internet, only few websites exploit the potential of combining user communities and online marketplaces. The reason is that users do not know which other users to trust, which makes them suspicious of engaging in online business, in particular if many unknown other parties are involved. This situation, however, can be alleviated by developing trust metrics such that a user can assess and identify trustworthy users. In the present study, we focus on developing a trust metric for estimating the trust of a target agent, who is unknown, through the information acquired from a group of advisor agents who had direct experience with the target agent, subject to possible trust uncertainty.

Each entity in a social network can be represented as an agent who is interacting with its network of trustees, which we refer to as advisors, where each advisor agent in turn is in interaction with an agent of interest, which we refer to as a target agent. Each interaction can be considered as a trust evaluation between the trustor agent, i.e., the agent who trusts another entity, and the trustee agent, i.e., the agent whom is being

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 283–299, 2013.

trusted. In the context of interactions between a service provider (trustee) and customers (trustors), some companies (e.g., e-bay and amazon) provide means for their customers to provide their feedback on the quality of the services they receive, under the form of a rating chosen out of a finite set of discrete values. This leads to a multi-valued domain of trust, where each trust rating represents the level of trustworthiness of the trustor agent as viewed by the trustee agent. While most of the web applications ask users to provide their feedbacks within such a multi-valued rating domain, most studies [1], [2], [3] and [4] are restricted to binary domains. Hence, our motivation for developing a multi-valued trust domain where each agent can be evaluated within a multi-valued set of ratings.

An agent may ask its advisors to provide information on a target agent who is unknown to him. The advisors are not necessarily truthful (e.g., competition among market shares, medical records when buying a life insurance) and therefore may manipulate their information before reporting it. In addition, the advisor agents' trustworthy behaviour may differ from one interaction to another, leading to some uncertainty about the advisors' trustworthiness and the accuracy of information revealed by them.

Possibility distribution is a flexible tool for modelling an agent's trust considering uncertainties which arises form either lack of sufficient information about an agent's trust or the variability in the agent's degree of trustworthiness. Possibility theory was first introduced by [5] and further developed by Dubois and Prade [6]. It was utilized, e.g., to model reliability [7]. We use possibility distributions to represent the trust of an agent with respect to its uncertainties. Further, we propose an approach on merging the possibility distributions of an explorer agent's trust in it's advisors with the reported possibility distributions by the advisors on a target agent's trust. The resulted possibility distribution represents an estimation of the explorer's trust in the target agent. The rest of the paper is structured as follows: Section 2 describes the related works. Section 3 provides a detailed description of the problem domain. Section 4 reviews some fusion rules for merging possibility distributions. In Section 6, we propose our approach on merging two different sets of the possibility distributions in order to estimate the target agent's trust. Finally, in Section 7, extensive experimental evaluations are presented to validate the proposed merging approach.

2 Related Work

Considerable research has been accomplished in multi-agent systems providing models of trust and reputation, a detailed overview of which is provided in [8]. In reputation models, an aggregation of opinions of members towards an individual member which is usually shared among those members is maintained. Starting with [9], the reputation of an agent can be evaluated and updated by agents over time. However, it is implicitly assumed that the agent's trust is a fixed unknown value at each specific time which does not capture the uncertainties in an agent's trust. Regret [10] is another reputation model which describes different dimensions of reputation (e.g. "individual dimension", "social dimension"). However, in this model the manipulation of information and how it can be handled is not addressed. Some trust models try to capture different dimensions of trust. In [11] a multi-dimensional trust containing elements like success, cost, timelines and quality is presented. The focus in this work is on the possible criteria that is required to build a trust model. However, the uncertainty in an agent's behavior and how it can be captured is not considered.

The work of [12] estimates the trust of an agent considering "direct experience", "witness information", "role-based rules" and "third-party references provided by the target agents". Although the latter 2 aspects are not included in our model, it is based on the assumption that the agents are honest in exchanging information with one another. In addition, despite the fact that the underlying trust of an agent is assumed to have a normal distribution, the estimated trust is a single value instead of a distribution. In other words, it does not try to measure the uncertainty associated with the occurrence of each outcome of the domain considering the results of the empirical experiments.

In all of the above works the uncertainty in the trust of an agent is not considered. We now review the works that address uncertainty. Reece *et al.* [3] present a multi-dimensional trust in which each dimension is binary (successful or unsuccessful) and corresponds to a service provided in a contract (video, audio, data service, etc.). This paper is mainly concentrated on fusing information received from agents who had direct observations over a subset of services (incomplete information) to derive the complete information on the target entity while our work focuses on having an accurate estimation when there is manipulation in the acquired information.

Yu and Singh [13] measure the probability of trust, distrust and uncertainty of an agent based on the outcomes of inter-agent interactions. The uncertainty measured in this work is equal to the frequency of the interaction results in which the agent's performance is neither highly trustworthy nor highly untrustworthy which can be inferred as lack of both trust and distrust in the agent. However, the uncertainty that we capture is due to lack of adequate information on an agent's trust or the variability in the agent's degree of trustworthiness regardless of how trustworthy the agent is. In other words, when an agent acts with high uncertainty it's degree of trustworthiness is hard to predict for future interactions. We do not consider uncertainty as lack of trust or distrust, but the variability in the degree of trustworthiness. In both works of [13] and [3] the possibility of having malicious agents providing falsified reports is ignored. The works of [1] and [2] provide probabilistic computational models measuring belief, disbelief and uncertainty from binary interactions (positive or negative). Although the manipulation of information by the reporter agents is not considered in these works, they split the interval of $[0, 1]$ between these 3 elements measuring a single value for each one of them. We do not capture uncertainty in the same sense by measuring a single value, instead we consider uncertainty by measuring the likelihood of occurrence of every trust element in the domain and therefore capturing the possible deviation in the degree of trustworthiness of the agents.

One of the closest works to our model which includes both uncertainty and the manipulation of information is Travos [4]. Although this work has a strong probabilistic approach and covers many issues, it is yet restricted to binary domain of events where each interaction, which is driven from the underlying probability that an agent fulfills it's obligations, is either successful or unsuccessful. Our work is a generalization of this work in the sense that it is extended to a multi-valued domain where we associate a probability to the occurrence of each trust value in the domain. Extension of the Travos model from binary to multi-valued event in the probabilistic domain is quite challenging due to its technical complexity. We use possibility theory which is a flexible and strong tool to address uncertainty and at the same time it is applicable to multi-valued domains. BLADE [14] is another recent trust model which considers subjectivity, deception and change. While we address the latter two features, the first one is missing in our model. The subjectivity reflects the personal point of view of each agent regarding the quality of a service, which can differ from an agent to another. BLADE considers the uncertainty arising from variability, however, it does not model the uncertainty due to ignorance. This is a shortcoming of the probabilistic models which are too normative to consider all sorts of uncertainty [6] and cannot demonstrate the uncertainty arising from ignorance .

3 Multi-agent Platform

In this section, we present the components that build the multi-agent environment and the motivation behind each choice. We first discuss the set of trust values (Section 3.1), the agent's internal trust distribution (Section 3.2) and the interactions among the agents (Section 3.3). Later, we describe the formation of the possibility distribution on an agent's trust (Section 3.4) and the possible agent information manipulations (Section 3.5). Finally, the game scenario in this paper is discussed (Section 3.6).

3.1 Trust Values

Service providers ask customers to provide their feedbacks on the received services commonly in form of a rating selected from a multi-valued set. The selected rating indicates a customer's degree of satisfaction or, in other words, its degree of trust in the provider's service. This motivates us to consider a multi-valued trust domain. We define a discrete multi-valued set of trust ratings denoted by T, with $\underline{\tau}$ being the lowest, $\overline{\tau}$ being the highest and $|T|$ representing the number of trust ratings. All trust ratings are within $[0, 1]$ and they can take any value in this range. However, if the trust ratings are distributed in equal intervals, the ith trust rating equals to: $(i - 1)/(|T| - 1)$ for $i = 1, 2, \ldots, |T|$. For example, if $|T| = 5$, then the set of trust ratings is $\{0, 0.25, 0.5, 0.75, 1\}$.

3.2 Internal Probability Distribution of an Agent's Trust

In our multi-agent platform, each agent is associated with an internal probability distribution of trust, which is only known to the agent. This allows modelling a specific degree of trustworthiness in that agent where each trust rating τ is given a probability of occurrence. In order to model a distribution, given its minimum, maximum, peak, degree of skewness and peakness, we use a form of beta distribution called modified pert distribution [15]. It can be replaced by any distribution that provides the above mentioned parameters. Well known distributions, e.g., normal distribution, are not employed as they do not allow positive or negative skewness of the distribution. In modified pert distribution, the peak of the distribution, which is denoted by τ_a^{PEAK}, has the highest probability of occurrence. This means that while the predominant behavior of the agent is driven by τ_a^{PEAK} and the trust ratings next to it, there is a small probability that the agent does not follow its dominant behavior. Figure 1(a) demonstrates an example of the internal trust distribution of an agent. The closer τ_a^{PEAK} is to $\overline{\tau}$, the more trustworthy the agent is and vice-versa.

3.3 Interaction between Agents

When a customer rates a provider's service, its rating depends not only on the provider's quality of service but also on the customer's personal point of view. In this paper, we just model the provider's quality of service. In each interaction a trustor agent, say α, requests a service from a trustee agent, say β. Agent β should provide a service in correspondence with its degree of trustworthiness which is implied in its internal trust distribution. On this purpose, it generates a random value from the domain of T by using its internal probability distribution of trust. The peak of the internal trust distribution, τ_a^{PEAK}, has the highest probability of selection while other trust ratings in T have a relatively smaller probability to be chosen. This will produce a mostly specific and yet not deterministic value. Agent β reports the generated value to α which represents the quality of service of β in that interaction.

3.4 Building Possibility Distribution of Trust

Possibility theory is one of the current theories for addressing uncertainty arising from variability and ignorance. Variability is due to the fluxing behaviour the system under study while ignorance is due to lack of sufficient information about the system under study. Probability distributions are too normative to address all sorts of uncertainty [6]. It can address variability [16], [17], but not ignorance. Ignorance can be represented by interval analysis or possibility theory [6]. We use possibility theory as it is capable of addressing both types of uncertainty. Moreover, as mentioned in [16], it is the simplest theory for addressing incomplete information (ignorance). These are the main factors guiding us to use the possibility theory for addressing uncertainty in this research.

Once a number of interactions between a trustor agent, α, and a trustee agent, β, agent α is completed, we can model the trust distribution of β, by usage of the values received from β during their interactions. If the number of interactions between the agents is high enough, the frequencies of each trust rating can almost represent the internal trust distribution of β. Otherwise, if few interactions are made, the randomly generated values may not represent the underlying distribution of β's trust [18]. In order to model an agent α's trust with respect to the uncertainty associated with the occurrence of each trust rating in the domain, we use possibility distributions which can present the degree of possibility of each trust rating in T. A possibility distribution is defined as: $\Pi : T \rightarrow [0, 1]$ with $\max\limits_{\tau \in T} \Pi(\tau) = 1$.

We apply the approach of [18] to build a possibility distribution from empirical data given the desired confidence level. In this approach, first simultaneous confidence intervals for all trust ratings in the domain are measured by usage of the empirical data (which in our model are derived from interaction among agents). The measured confidence intervals are based on the research presented in [19]. Then, an algorithm is proposed to simplify the process of finding the maximum value of the enumerators number of optimizations that are required to take in order to measure the possibility of each trust rating τ. Due to lack of space we cannot elaborate more on this process. The detailed procedure is presented in [18].

3.5 Manipulation of the Possibility Distributions

An agent, say a^{S}, needs to acquire information about the degree of trustworthiness of agent a^{D} unknown to him. On this purpose, it acquires information from its advisors like a who are known to a^{S} and have already interacted with a^{D}. Each agent a is not necessarily truthful for reasons of self-interest, therefore it may manipulate the possibility distribution it has built on a^{D}'s trust before reporting it to a^{S}. The degree of manipulation of the information by agent a is based on its internal probability distribution of trust. More specifically, if the internal trust distribution of agents a and a' indicate that a's degree of trustworthiness is lower than a', then the reported possibility distribution of a is more prone to error than a'. The following two algorithm introduced in this section are examples of manipulation algorithms:

Algorithm I

```
for each $\tau \in T$ do
    $\tau' \leftarrow$ random trust rating from $T$, according
        to agent $a$'s internal trust distribution
    error$_\tau = 1 - \tau'$
    $\Pi_{a \to a^{\text{D}}}(\tau) = \widehat{\Pi}_{a \to a^{\text{D}}}(\tau) +$ error$_\tau$
end for
```

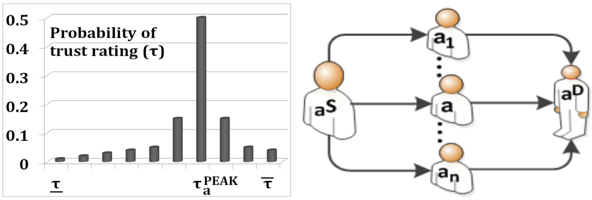
where $\widehat{\Pi}_{a \to a^{\text{D}}}(\tau), \forall \tau \in T$ is the possibility distribution built by a through its interactions with a^{D} and $\Pi_{a \to a^{\text{D}}}(\tau), \forall \tau \in T$ is the manipulated possibility distribution which is reported to a^{S}. In this algorithm, for each trust rating $\tau \in T$ a random trust value, τ', is generated following the internal trust distribution of agent a. For highly trustworthy agents, the randomly generated value of τ' is closer to $\overline{\tau}$ and the subsequent error (error$_\tau$) is closer to 0. Therefore the manipulation of the possibility value of $\widehat{\Pi}_{a \to a^{\text{D}}}(\tau)$, is insignificant. On the other hand, for highly untrustworthy agents, the value of τ' is closer to $\underline{\tau}$ and therefore the derived error, error$_\tau$, is closer to 1. In such a case, the possibility value of $\widehat{\Pi}_{a \to a^{\text{D}}}(\tau)$ is considerably modified causing noticeable change in the original values.

After measuring the distribution of $\Pi_{a \to a^{\text{D}}}(\tau), \forall \tau \in T$, it is normalized and then reported to a^{S}. The normalization satisfies: (1) the possibility value of every trust rating τ in T is in $[0, 1]$, and (2) the possibility value of at least one trust rating in T equals to 1. let $\widetilde{\Pi}(\tau)$ be a non-normalized possibility distribution. Either of the following formulas [7] generates a normalized possibility distribution of $\overline{\Pi}(\tau)$:

$$(1)\ \overline{\Pi}(\tau) = \widetilde{\Pi}(\tau)/h, \tag{1}$$

$$(2)\ \overline{\Pi}(\tau) = \widetilde{\Pi}(\tau) + 1 - h, \tag{2}$$

$$\text{where } h = \max_{\tau \in T} \widetilde{\Pi}(\tau).$$



(a) Internal Prob. Dist. of a (b) Network of Agents

Fig. 1. Multi-Agent Platform

The second manipulation algorithm is the following:

Algorithm II

```
for each $\tau \in T$ do
    $\tau' \leftarrow$ random trust rating from $T$, according
            to agent $a$'s internal trust distribution
    $\max\_\text{error}_\tau = 1 - \tau'$
    $\text{error}_\tau$ = random value in $[0, \max\_\text{error}_\tau]$
    $\Pi_{a \to a^{\text{D}}}(\tau) = \widehat{\Pi}_{a \to a^{\text{D}}}(\tau) + \text{error}_\tau$
end for
```

As for algorithm I, the distribution of $\Pi_{a \to a^{\text{D}}}(\tau), \forall \tau \in T$ is normalized before being reported to a^{S}. In algorithm II, an additional random selection value is added where the random value is selected uniformly in $[0, \max\_\text{error}_\tau]$. In algorithm I, the trust rating of τ_a^{PEAK} and the trust values next to it have a high probability of being selected. The error added to $\widehat{\Pi}_{a \to a^{\text{D}}}(\tau)$ may be neglected when the distribution is normalized. However, in algorithm II, if an agent is highly untrustworthy the random trust value of τ' is close to $\underline{\tau}$ and thereupon the error value of $\max\_\text{error}_\tau$ is close to 1. This causes the uniformly generated value in $[0, \max\_\text{error}_\tau]$ considerably random and unpredictable which makes the derived possibility distribution highly erroneous after normalization. On the other hand, if an agent is highly trustworthy, the error value of $\max\_\text{error}_\tau$ is close to zero and the random value generated in $[0, \max\_\text{error}_\tau]$ would be even smaller, making the error of the final possibility distribution insignificant. While incorporating some random process, both algorithms manipulate the possibility distribution based on the agent's degree of trustworthiness causing the scale of manipulation by more trustworthy agents smaller and vice-versa. However, the second algorithm acts more randomly. We provide these algorithms to observe the extent of dependency of the derived results with respect to a specific manipulation algorithm employed.

3.6 Game Scenario

In this paper, we study a model arising in social networks where agent a^{S} makes a number of interactions with each agent a in a set $A = \{a_1, a_2, \ldots, a_n\}$ of n agents (agent a^{S}'s advisors), assuming each agent $a \in A$ has carried out some interactions with agent a^{D}. Agent a^{S} builds a possibility distribution of trust for each agent in A by usage of the empirical data derived throughout their interactions, as described in Section 3.4. Each agent in A, in turn, builds an independent possibility distribution of trust through its own interactions with agent a^{D}. When a^{S} wants to evaluate the level of trustworthiness of a^{D}, who is unknown to him, it acquires information from its advisors, A, to report their measured possibility distributions on a^{D}'s trust. Agents in A are not necessarily truthful. Therefore, through usage of the manipulation algorithms, they manipulate their own possibility distributions of $\widehat{\Pi}_{a \to a^{\mathrm{D}}}(\tau), \forall \tau \in T$ in correspondence with their degree of trustworthiness and report the manipulated distributions to a^{S}. Agent a^{S} uses the reported distributions of $\Pi_{a \to a^{\mathrm{D}}}(\tau), \forall \tau \in T$ by each agent $a \in A$ and its trust distribution in agent a, represented by $\Pi_{a^{\mathrm{S}} \to a}(\tau), \forall \tau \in T$, in order to estimates the possibility distribution of a^{D}'s trust. Figure 1(b) illustrates the connections between the agents in this paper. The head of each arrow, connecting two agents, indicates the trustor while the tail of the arrow indicates the trustee in a trust relation.

4 Fusion Rules Considering the Trust of the Agents

Let $\tau_{a^{\mathrm{S}} \to a} \in [0,1]$ be a single trust value of agent a^{S} in agent a and $\Pi_{a \to a^{\mathrm{D}}}(\tau), \forall \tau \in T$ represent the possibility distribution of agent a's trust in agent a^{D} as reported by a to a^{S}. We now look at different fusion rules for merging the possibility distributions of $\Pi_{a \to a^{\mathrm{D}}}(\tau), \forall \tau \in T, \forall a \in A$, with respect to the trust values of $\tau_{a^{\mathrm{S}} \to a}, \forall a \in A$, in order to get a possibility distribution of $\Pi_{a^{\mathrm{S}} \to a^{\mathrm{D}}}(\tau), \forall \tau \in T$, representing a^{S}'s trust in a^{D}. We explore three most commonly used fusion rules. The first one is the Trade-off (To) rule [20], which builds a weighted mean of the possibility distributions:

$$\Pi^{To}_{a^{\mathrm{S}} \to a^{\mathrm{D}}}(\tau) = \sum_{a \in A} \omega_a \times \Pi_{a \to a^{\mathrm{D}}}(\tau), \tag{3}$$

where $\omega_a = \tau_{a^{\mathrm{S}} \to a} / \sum\limits_{a \in A} \tau_{a^{\mathrm{S}} \to a}$ for $\tau \in T$, and $\Pi^{To}_{a^{\mathrm{S}} \to a^{\mathrm{D}}}(\tau), \forall \tau \in T$ indicates the trust distribution of a^{S} in a^{D}, measured by Trade-off rule. Note that the trade-off rule considers all of the possibility distributions reported by the agents in A. However, the degree of influence of the possibility distribution of $\Pi_{a \to a^{\mathrm{D}}}(\tau), \forall \tau \in T$ is weighted by the normalized trust of agent a^{S} in agent a (which is ω_a).

The next two fusion rules belong to a family of rules which modify the possibility distribution of $\Pi_{a \to a^{\mathrm{D}}}(\tau), \forall \tau \in T$ based on the trust value associated with it, $\tau_{a^{\mathrm{S}} \to a}$, and then take an intersection [21] of the modified distributions. We refer to this group of fusion rules as Trust Modified (TM) rules. Therein, if $\tau_{a^{\mathrm{S}} \to a} = 1$, $\Pi_{a \to a^{\mathrm{D}}}(\tau), \forall \tau \in T$ remains unchanged, meaning that agent a^{S}'s full trust in a results in total acceptance of possibility distribution of $\Pi_{a \to a^{\mathrm{D}}}(\tau), \forall \tau \in T$ reported by a. The less agent a is trustworthy, the less its reported distribution is reliable and consequently its reported distribution of $\Pi_{a \to a^{\mathrm{D}}}(\tau), \forall \tau \in T$ is moved closer towards a uniform distribution by TM rules. In the context of possibility distributions, the uniform distribution provides no information as all trust values in domain T are equally possible which is referred to as complete ignorance [22]. The more a distribution of $\Pi_{a \to a^{\mathrm{D}}}(\tau), \forall \tau \in T$ gets closer to a uniform distribution, the less likely it is selected in the intersection phase [21] of the TM rules. We selected the following two TM fusion rules:

Yager [23]:

$$\mathit{\Pi}^{Y}_{a^{\mathrm{S}}\to a^{\mathrm{D}}}(\tau) = \min_{a\in A}\left[\tau_{a^{\mathrm{S}}\to a} \times \mathit{\Pi}_{a\to a^{\mathrm{D}}}(\tau) + 1 - \tau_{a^{\mathrm{S}}\to a}\right]. \tag{4}$$

Dubois and Prade [24]:

$$\mathit{\Pi}^{DP}_{a^{\mathrm{S}}\to a^{\mathrm{D}}}(\tau) = \min_{a\in A}\left[\max(\mathit{\Pi}_{a\to a^{\mathrm{D}}}(\tau), 1 - \tau_{a^{\mathrm{S}}\to a})\right]. \tag{5}$$

In Yager's fusion rule, the possibility of each trust value τ moves towards a uniform distribution as much as $(1-\tau_{a^{\mathrm{S}}\to a})$, which is the extent to which the agent a is not trusted. In Dubois and Prade's fusion rule, when an agent's trust declines, the $\max$ operator would more likely select $1-\tau_{a^{\mathrm{S}}\to a}$ and, hence, the information in $\mathit{\Pi}_{a\to a^{\mathrm{D}}}(\tau), \forall\tau\in T$ reported by a gets closer to a uniform distribution. Consequently, it has a less chance of being selected by the min operator.

Once a fusion rule in this Section is applied, the resulted possibility distribution of $\mathit{\Pi}_{a^{\mathrm{S}}\to a^{\mathrm{D}}}(\tau), \forall\tau\in T$ is normalized to represent the possibility distribution of agent a^{S}'s trust in a^{D}.

5 Merging Successive Possibility Distributions

In this section, we present the main contribution, i.e., a methodology for merging the possibility distribution of $\mathit{\Pi}_{a^{\mathrm{S}}\to a}(\tau), \forall\tau\in T, \forall a\in A$ (representing the trust of agent a^{S} in its advisors) with the possibility distribution of $\mathit{\Pi}_{a\to a^{\mathrm{D}}}(\tau), \forall\tau\in T, \forall a\in A$ (representing the trust of the agent set A in agent a^{D}). These two possibility distributions are associated to the trust of entities at successive levels in a multi-agents systems and hence giving it such a name.

In order to perform such a merging, we need to know how the distribution of $\mathit{\Pi}_{a\to a^{\mathrm{D}}}(\tau), \forall\tau\in T$ changes, depending on the characteristics of the possibility distribution of $\mathit{\Pi}_{a^{\mathrm{S}}\to a}(\tau), \forall\tau\in T$. We distinguish the following cases for a proper merging of the successive possibility distributions.

5.1 Specific Case

Consider a scenario where

$$\exists!\tau'\ ,\underline{\tau}\leq\tau'\leq\overline{\tau} \text{ and } \mathit{\Pi}_{a^{\mathrm{S}}\to a}(\tau) = \begin{cases}1, \tau=\tau'\\ \text{0, otherwise}\end{cases}$$

i.e., only one trust value is possible in the domain of T and the possibility of all other trust values is equal to 0. Then, trust of agent a^{S} in agent a can be associated with a single value of $\tau_{a^{\mathrm{S}}\to a}=\tau'$ and the fusion rules described in section 4 can be applied to get the possibility distribution of $\mathit{\Pi}_{a^{\mathrm{S}}\to a^{\mathrm{D}}}(\tau), \forall\tau\in T$.

Considering the TM fusion rules, for each agent a, first the possibility distribution of $\mathit{\Pi}_{a\to a^{\mathrm{D}}}(\tau), \forall\tau\in T$ is transformed based on the trust value of $\tau_{a^{\mathrm{S}}\to a}=\tau'$ as discussed in Section 4. Then, an intersection of the transformed possibility distribution is taken and the resulted distribution is normalized to get the possibility distribution of $\mathit{\Pi}_{a^{\mathrm{S}}\to a^{\mathrm{D}}}(\tau), \forall\tau\in T$.

5.2 General Case

For each agent a, we have a subset of trust ratings, which we refer to as T_a^{Pos}, such that:

$$\begin{aligned} &1) \quad T_a^{\text{Pos}} \subset T, \\ &2) \quad \text{If } \Pi_{a^{\text{S}} \to a}(\tau) > 0, \text{ then } \tau \in T_a^{\text{Pos}}, \\ &3) \quad \text{If } \Pi_{a^{\text{S}} \to a}(\tau) = 0, \text{ then } \tau \in \{T - T_a^{\text{Pos}}\}. \end{aligned}$$

Each trust rating value in T_a^{Pos} is possible. This means that the trust of agent a^{S} in a can possibly take any value in T_a^{Pos} and consequently any trust rating $\tau \in T_a^{\text{Pos}}$ can be possibly associated with $\tau_{a^{\text{S}} \to a}$. However, the higher the value of $\Pi_{a^{\text{S}} \to a}(\tau)$, the higher the likelihood of occurrence of trust rating $\tau \in T_a^{\text{Pos}}$. We use the possibility distribution of $\Pi_{a^{\text{S}} \to a}(\tau), \forall \tau \in T$ to get the relative chance of happening of each trust rating in T_a^{Pos}. In this approach, we give each trust rating τ, a Possibility Weight (PW) equal to:

$$PW(\tau) = \Pi_{a^{\text{S}} \to a}(\tau) / \sum_{\tau' \in T_a^{\text{Pos}}} \Pi_{a^{\text{S}} \to a(\tau')}. \tag{6}$$

Higher value of $PW(\tau)$ implies more occurrences chance of the trust rating τ. Hence, any trust rating $\tau \in T_a^{\text{Pos}}$ is possible to be observed with a weight of $PW(\tau)$ and merged with $\Pi_{a \to a^{\text{D}}}(\tau), \forall \tau \in T$ using one of the fusion rules.

Considering the General Case, there are a total of $|A| = n$ agents and each agent a has a total of $|T_a^{\text{Pos}}|$ possible trust values. For a possible estimation of $\Pi_{a^{\text{S}} \to a^{\text{D}}}(\tau), \forall \tau \in T$, we need to choose one trust rating of $\tau \in T_a^{\text{Pos}}$ for each agent $a \in A$. Having $|A| = n$ agents and a total of $|T_a^{\text{Pos}}|$ possible trust ratings for each agent $a \in A$, we can generate a total of $\prod_{a \in A} |T_a^{\text{Pos}}| = K$ possible ways of getting the final possibility of $\Pi_{a^{\text{S}} \to a^{\text{D}}}(\tau), \forall \tau \in T$. This means that any distribution out of K distributions is possible. However, they are not equally likely to happen. If agent a^{S} chooses trust rating $\tau_1 \in T_{a_1}^{\text{Pos}}$ for agent a_1, $\tau_2 \in T_{a_2}^{\text{Pos}}$ for agent a_2, and finally τ_n for agent $a_n \in T_{a_n}^{\text{Pos}}$, then the possibility distribution of $\Pi_{a^{\text{S}} \to a^{\text{D}}}(\tau), \forall \tau \in T$ derived from these trust ratings has an Occurrence Probability(OP) of $\prod_{i=1}^{n} PW(\tau_i)$.

For every agent a, we have: $\sum_{\tau \in T_a^{\text{Pos}}} PW(\tau) = 1$, then considering all agents we have:

$$\sum_{\tau_1 \in T_{a_1}^{\text{Pos}}} \sum_{\tau_2 \in T_{a_2}^{\text{Pos}}} \cdots \sum_{\tau_n \in T_{a_n}^{\text{Pos}}} PW(\tau_1) \times PW(\tau_2) \times \ldots \times PW(\tau_n) = 1. \tag{7}$$

As can be observed above, the PW is normalized in such a way that, for every set of trust ratings $\{\tau_1, \tau_2, \ldots, \tau_n\}$ (where $\tau_i \in T_{a_i}^{\text{Pos}}$), the corresponding OP of this set can be measured through multiplication of PW of the trust ratings in the set, namely $PW(\tau_1) \times PW(\tau_2) \times \ldots \times PW(\tau_n)$.

5.3 Trust Event Coefficient

The $PW(\tau)$ value shows the relative possibility of τ compared to other values in T of an agent a. However, we still need to compare the possibility of a given trust rating τ, for an agent a, compared to other agents in A. If the possibility weights of two agents are equal, say 0.2 and 0.8 for trust ratings $\underline{\tau}$ and $\overline{\tau}$, and the number of interactions with the first agent is much higher than the second agent, we should give more credit to the first agent's reported distribution of $\Pi_{a \to a^{\mathrm{D}}(\tau)}, \forall \tau \in T$. However, the current model is unable of doing so. Therefore, we propose to use a Trust Event Coefficient for each trust value τ, denoted by $\textsc{tec}(\tau)$, in order to consider the number of interactions, which satisfies:

1) If $m_\tau = 0,\ \textsc{tec}(\tau) = 0$
2) If $\Pi_{a^{\mathrm{S}} \to a}(\tau) = 0,\ \textsc{tec}(\tau) = 0$
3) If $m_\tau \geq m_{\tau'},\ \textsc{tec}(\tau) \geq \textsc{tec}(\tau')$
4) If $m_\tau = m_{\tau'}$ and $\Pi_{a^{\mathrm{S}} \to a}(\tau) \geq \Pi_{a^{\mathrm{S}} \to a}(\tau')$,
$\textsc{tec}(\tau) \geq \textsc{tec}(\tau')$,

where $\tau \in T_a^{\mathrm{Pos}}$, m_τ is the frequency of the trust rating τ in the interactions between agents a^{S} and a. Considering conditions 1) and 2), if the number of occurrences of trust rating τ or its corresponding possibility is 0, then $\textsc{tec}(\tau)$ is also zero. Condition 3) increases the value of $\textsc{tec}$ by increasing the number of occurrences of trust rating τ. As observed in Condition 4), if the number of observances of two trust ratings, τ and τ' are equal, then the trust rating with higher possibility is given the priority. When comparing the number of interactions and the possibility value of $\Pi_{a^{\mathrm{S}} \to a}(\tau)$, the priority is given first to number of the interactions, and then, to the the possibility value of $\Pi_{a^{\mathrm{S}} \to a}(\tau)$ in order to avoid giving preference to the possibility values driven out of few interactions. The following formula is an example of a $\textsc{tec}$ function which satisfies the above conditions.

$$\textsc{tec}(\tau) = \begin{cases} 0, m_\tau = 0 \text{ or } \Pi_{a^{\mathrm{S}} \to a}(\tau) = 0 \\ [1/(\gamma \times m_\tau)]^{(1/m_\tau)} + \frac{\Pi_{a^{\mathrm{S}} \to a}(\tau)}{\chi}, \text{ otherwise} \end{cases} \tag{8}$$

where $\gamma > 1$ is the discount factor and $\chi \gg 1$. Higher values of γ impede the convergence of $\textsc{tec}(\tau)$ to one and vice-versa. χ which is a very large value insures that the influence of $\Pi_{a^{\mathrm{S}} \to a}(\tau)$ on $\textsc{tec}(\tau)$ remains trivial and is noticeable only when the number of interactions are equal. In this formula, as m_τ grows, $\textsc{tec}(\tau)$ converges to one. $\textsc{tec}(\tau)$ can be utilized as a coefficient for trust rating τ when comparing different agents. Note that the General Case mentioned above gives the guidelines for merging successive possibility distributions and $\textsc{tec}$ feature is only used as an attribute when the number of interactions should be considered and can be ignored otherwise.

6 Possibility Distribution of Agent a^S's Trust in Agent a^D

We propose two approaches for estimating the final possibility distribution of $\Pi_{a^{\mathrm{S}} \to a^{\mathrm{D}}}(\tau), \forall \tau \in T$ considering different available possible choices. The first approach is to consider all K possibility distributions of $\Pi_{a^{\mathrm{S}} \to a^{\mathrm{D}}}(\tau), \forall \tau \in T$ and take the weighted mean of them by giving each distribution $\Pi_{a^{\mathrm{S}} \to a^{\mathrm{D}}}(\tau), \forall \tau \in T$ a weight equal to its Occurrence Probability (OP), measured by multiplying the possibility weight of the trust values, $PW(\tau_i)$, that are used to build $\Pi_{a^{\text{S}} \to a^{\text{D}}}(\tau), \forall \tau \in T$.

In the second approach, we only consider the trust ratings, $\tau \in T$ such that $\Pi_{a^{\text{S}} \to a}(\tau) = 1$. In other words, we only consider the trust ratings that have the highest weight of PW in the T_a^{Pos} set. Consequently, the $\Pi_{a^{\text{S}} \to a^{\text{D}}}(\tau), \forall \tau \in T$ distributions derived from these trust values have the highest OP value which makes them the most expected distributions. We denote by μ_a the number of trust ratings, $\tau \in T_a^{\text{Pos}}$ that satisfy $\Pi_{a^{\text{S}} \to a}(\tau) = 1$ for agent a. In this approach, we only select the trust ratings in μ_a for each agent $a \in A$ and build the possibility distributions of $\Pi_{a^{\text{S}} \to a^{\text{D}}}(\tau), \forall \tau \in T$ out of those trust ratings. After building $M = \prod_{a \in A} \mu_a$ different possibility distributions of $\Pi_{a^{\text{S}} \to a^{\text{D}}}(\tau)$, we compute their average, since all of them have equal OP weight.

Proposition 1. *In both approaches, the conditions of the general case described in the previous section are satisfied.*

Proof. Proof can be easily done by enumerating different cases.

Due to the computational burden of the first approach (which requires building K distributions of $\Pi_{a^{\text{S}} \to a^{\text{D}}}(\tau), \forall \tau \in T$), we used the second approach in our experiments as it only requires building M distributions.

To conclude this section, we would like to comment on the motivation behind using possibility distribution rather than probability distributions. Indeed, if probability distributions were used instead of possibility distributions, in order to consider uncertainty a confidence interval should be measured in place of each possibility value. Consequently, in order to represent the trust of agent a^{S} in an agent a, each possibility value $\Pi_{a^{\text{S}} \to a}(\tau)$ should be replaced with a confidence interval. The same representation should be applied for each agent $a \in A$'s trust in a^{D}. Now, in order to estimate the probability distribution of agent a^{D}'s trust with respect to its uncertainty, we need to find some tools for merging the confidence intervals of a^{S}'s trust in A with the confidence intervals of A's trust in a^{D}. To the best of our knowledge, no work addresses this issue, except for the following related works. In [25], the number of the occurrences of each element in the domain, which in our model is equivalent to the number of observance of each τ value in the interactions between agent a and a^{D}, is reported by agents in A to a^{S} and then, the confidence intervals on the trust of agent a^{D} is built by a^{S}. The work of [26] measures the confidence intervals of a^{D}'s trust out of several confidence intervals provided by agents in A. In both works, the manipulation of information by the agents in A is not considered and for building the confidence intervals of a^{D}, the trust of agent a^{S} in A is neglected. Despite lack of proper tool in the probability domain, we employed possibility distributions as they can address the same problem in a much simpler approach.

7 Experiments

We first introduce two metrics for evaluating the outcomes of our experiments and then present the experimental results.

7.1 Evaluation Metrics

Metric I - How Informative is a Possibility Distribution? In the context of the possibility theory, the uniform distribution contributes no information, as all of the trust ratings are equally possible and cannot be differentiated. This state is referred to as "complete ignorance" [22]. Consequently, the more a possibility distribution deviates from the uniform distribution, the more it contributes information. The following distribution provides the state of "complete knowledge" [22]:

$$\exists!\, \tau \in T : \Pi(\tau) = 1 \text{ and } \Pi(\tau') = 0,\ \forall \tau' \neq \tau, \tag{9}$$

where only one trust value in T has a possibility greater than 0. We assign an information level of 1 and 0 to the distribution of Eq.(9) and the uniform distribution, respectively. In the general case, the information level (denoted by I) of a distribution having a total of $|T|$ trust ratings, is equal to:

$$I(\Pi(\tau)) = \frac{1}{|T|-1} \sum_{\tau \in T} (1 - \Pi(\tau)). \tag{10}$$

Here the distance of each possibility value of $\Pi(\tau)$ from the uniform distribution is measured first for all trust ratings of T. Then, it is normalized by $|T| - 1$, since at least one trust rating must be equal to 1 (property of a possibility distribution).

Metric II - Estimated Error of a Possibility Distribution. In this section, we want to measure the difference between the estimated possibility distribution of agent a^{D}'s trust, as measured in Section 6, and the true possibility distribution of a^{D}'s trust. In order to measure the true possibility distribution of agent a^{D}'s trust, the true probability distribution of agent a^{D}'s trust (which is its internal probability distribution of trust) should be transformed to a possibility distribution. Dubois *et al.* [27] provide a probability to possibility transformation tool. Through usage of their tool, the true possibility distribution of a^{D}'s trust can be measured and then compared with the estimated distribution of $\Pi_{a^{\mathrm{S}} \to a^{\mathrm{D}}}(\tau), \forall \tau \in T$. Let $\Pi_{a^{\mathrm{S}} \to a^{\mathrm{D}}}(\tau), \forall \tau \in T$ denote an estimated distribution, as measured in Section 6, obtained from a fusion rule and let $\Pi_F(\tau), \forall \tau \in T$ represent the true possibility distribution of a^{D}'s trust, transformed from its internal probability distribution. The Estimated Error (EE) of $\Pi_{a^{\mathrm{S}} \to a^{\mathrm{D}}}(\tau), \forall \tau \in T$ is measured by taking the average of the absolute differences between the true and estimated possibility values over all trust ratings, $\tau \in T$. The EE metric is measured as:

$$EE(\Pi_F(\tau)) = \frac{1}{|T|} \sum_{\tau \in T} \left| \Pi_{a^{\mathrm{S}} \to a^{\mathrm{D}}}(\tau) - \Pi_F(\tau) \right|. \tag{11}$$

7.2 Experimental Results

We set up extensive experiments to evaluate our merging approaches. We divide the set A of agents into three subsets. Each subset simulates a specific level of trustworthiness in the agents. The subsets are: A^{FT} subset of Fully Trustworthy agents where the peak of the probability trust distribution is 1, A^{HT} subset of Half Trustworthy agents where the

Table 1. Agent distribution corresponding to x values in Figure 2

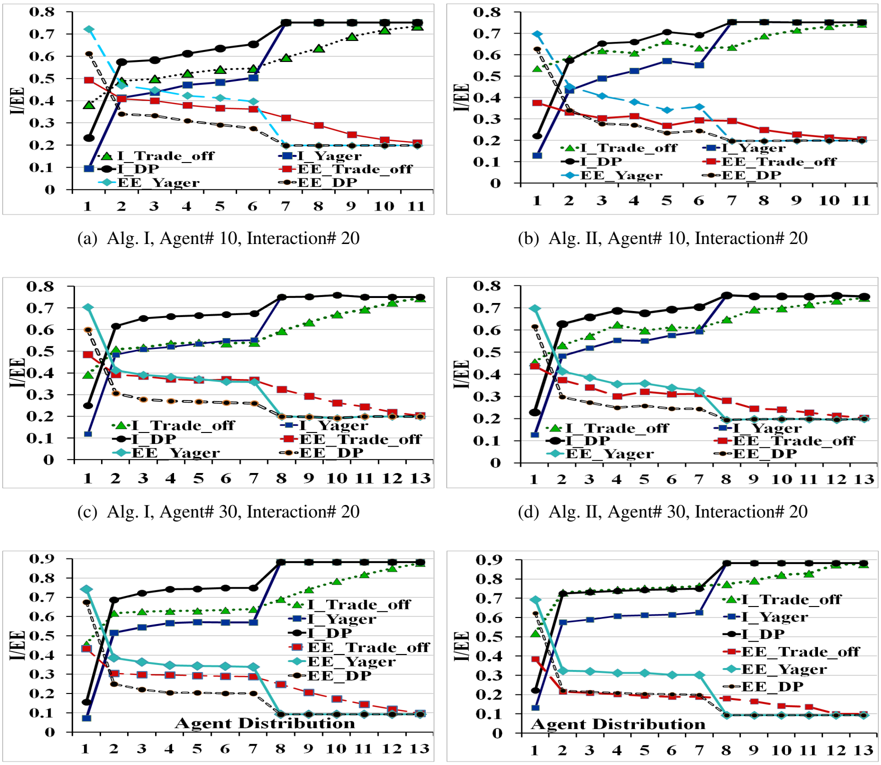
	Agent Distribution in (c) to (f)													Agent Distribution in (a) and (b)										
x	1	2	3	4	5	6	7	8	9	10	11	12	13	1	2	3	4	5	6	7	8	9	10	11
$\|A^{\mathrm{FT}}\|$	0	0	0	0	0	0	0	5	10	15	20	25	30	0	0	0	0	0	0	2	4	6	8	10
$\|A^{\mathrm{HT}}\|$	0	5	10	15	20	25	30	25	20	15	10	5	0	0	2	4	6	8	10	8	6	4	2	0
$\|A^{\mathrm{NT}}\|$	30	25	20	15	10	5	0	0	0	0	0	0	0	10	8	6	4	2	0	0	0	0	0	0

peak is 0.5 and A^{NT} subset of Not Trustworthy agents where the peak is 0. We start with $A = A^{\mathrm{NT}}$ and gradually move the agents from $A = A^{\mathrm{NT}}$ to $A = A^{\mathrm{HT}}$ such that we reach the state of $A = A^{\mathrm{HT}}$ where all the agents belong to A^{HT}. Later, we move agents from $A = A^{\mathrm{HT}}$ to $A = A^{\mathrm{FT}}$ and we finally end up with $A = A^{\mathrm{FT}}$. Over this transformation, the robustness of the estimated distribution of $\Pi_{a^{\mathrm{S}} \to a^{\mathrm{D}}}(\tau), \forall \tau \in T$ is evaluated with respect to the nature of trustworthiness of the agents. We carry out separate experiments by changing: (1) The number of agents in the set A, (2) The number of interactions between each pair of connected agents, as demonstrated in Figure 1(b), and (3) The manipulation Algorithm. We intend to observe the influence of each one of these components on the final estimated distribution of $\Pi_{a^{\mathrm{S}} \to a^{\mathrm{D}}}(\tau), \forall \tau \in T$. In all experiments, the number of trust ratings, $|T|$, is equal to 5 (a commonly used value in most surveys).

Experiments with Manipulation Algorithm I. In the first set of experiments, the manipulation algorithm I is used by agents in A. Diagrams on the left side of Figure 2 represent three different experiments where the number of agents in A and the number of interactions among every pair of connected agents, as illustrated in Figure 1(b), have changed. Table 1 gives the distribution of agents A into $A^{\mathrm{FT}} \cup A^{\mathrm{NT}} \cup A^{\mathrm{HT}}$ over x axis values in Figure 2. The three left side diagrams of Figure 2 demonstrate that through migration of the agents from A^{NT} to A^{HT} and later to A^{FT}, the I increases and EE decreases. This is a consequence of the increase in the accuracy of information provided by the agents in A as they become more trustworthy.

Comparing the three experiments on the left side of Figure 2, increase in the number of agents in Figures 2(c) compared to 2(a), does not improve the results over high values of x, where the number of the agents in the A^{FT} subset is non-zero. This indicates that as long as the quality of the information reported by the agents in A does not improve, increase in the number of the agents will not improve the estimated distribution of $\Pi_{a^{\mathrm{S}} \to a^{\mathrm{D}}}(\tau), \forall \tau \in T$. However, from $x = 2$ to the case where all agents are in A^{HT}, EE reduces and I increases. This indicates that if agents are not completely trustworthy, an increase in the number of agents increments the quality of the estimated distributions. Comparing Figures 2(c) and 2(e), increase in the number of interactions in-between the agents has increased I and decreased EE in Figure 2(e) which is a consequence of higher information exchanges between the agents. Thus, the possibility distributions built by the agents are derived from more information which enhances the estimation accuracy.

Experiments with Manipulation Algorithm II. We repeat the same experiments with manipulation algorithm II to observe the extent of influence of the chosen manipulation algorithm by the set A on the final distribution of $\Pi_{a^{\mathrm{S}} \to a^{\mathrm{D}}}(\tau), \forall \tau \in T$. The Diagrams



(a) Alg. I, Agent# 10, Interaction# 20

(b) Alg. II, Agent# 10, Interaction# 20

(c) Alg. I, Agent# 30, Interaction# 20

(d) Alg. II, Agent# 30, Interaction# 20

(e) Alg. I, Agent# 30, Interaction# 50

(f) Alg. II, Agent# 30, Interaction# 50

Fig. 2. Experiments with Algorithms I and II in Different Multi-Agent Settings

on the right side of Figure 2 represent the results of these experiments. These graphs demonstrate the same trends as algorithm I. However, more volatility is observed in the graphs of Algorithm II compared to Algorithm I as the graphs are not monotonically changing over the x axis. Indeed, this is a consequence of the increased randomization of manipulation algorithm II compared to algorithm I.

Comparing the fusion rules, DP outperforms other fusion rules in all experiments of Algorithm I and II. This is due to the fact that the DP rule is more categoric in its ignorance of the agents who are not trustworthy compared to the other two fusion rules. We performed additional experiments and the results show that through higher number of interactions among the agents, increase in the agents degree of trustworthiness, increase in the number of agents in A, and finally decrease in the number of trust ratings ($|T|$), the quality of estimation results enhances.

8 Conclusions

In this paper we analysed the properties of merging successive possibility distributions, representing the trust of the agents in successive levels of a multi-agent system. This is the first work that merges successive possibility distributions generated at different levels of a multi-agent system. We used this approach in order to estimate the trust of an unknown agent, through the information provided by a set of self-interested agents who are partially known to the explorer agent. The approach presented in this paper can be extended to estimate the trust of an agent of interest in the multi-agent systems, who can be reached through a set of sequentially connected agents. In addition, we addressed uncertainty in our trust model, arising from variability and ignorance in the empirical data that are generated from an unknown distribution, through usage of the possibility distributions. Finally, we applied the proposed merging approach in extensive experiments to validate our estimation results.

References

1. Jøsang, A.: A logic for uncertain probabilities. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 9, 279–311 (2001)
2. Wang, Y., Singh, M.: Evidence-based trust: A mathematical model geared for multiagent systems. ACM Trans. Auton. Adapt. Syst. 5, 14:1–14:28 (2010)
3. Reece, S., Roberts, S., Rogers, A., Jennings, N.R.: A multi-dimensional trust model for heterogeneous contract observations. In: Proc. of AAAI, pp. 128–135 (2007)
4. Teacy, W.T.L., Patel, J., Jennings, N.R., Luck, M.: Travos: Trust and reputation in the context of inaccurate information sources. Auton. Agent Multi-Agent Sys. 12, 183–198 (2006)
5. Zadeh, L.: Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems 1, 3–28 (1978)
6. Dubois, D., Prade, H.: Possibility Theory: An Approach to the Computerized Processing of Uncertainty. Plenum Press (1988)
7. Delmotte, F., Borne, P.: Modeling of reliability with possibility theory. IEEE Transactions on Systems, Man, and Cybernetics 28, 78–88 (1998)
8. Ramchurn, S.D., Huynh, D., Jennings, N.R.: Trust in multi-agent systems. The Knowledge Engineering Review 19, 1–25 (2004)
9. Zacharia, G., Moukas, A., Maes, P.: Collaborative reputation mechanisms for electronic marketplaces. Decision Support Systems 29, 371–388 (2000)
10. Sabater, J., Sierra, C.: Regret: A reputation model for gregarious societies. In: 4th Workshop on Deception, Fraud and Trust in Agent Societies, pp. 61–69. ACM (2001)
11. Griffiths, N.: Task delegation using experience-based multi-dimensional trust. In: Proc. of AAMAS, pp. 489–496 (2005)
12. Huynh, T.D., Jennings, N.R., Shadbolt, N.R.: An integrated trust and reputation model for open multi-agent systems. In: Proc. of AAMAS, vol. 13, pp. 119–154 (2006)
13. Yu, B., Singh, M.P.: An evidential model of distributed reputation management. In: Proc. of AAMAS, pp. 294–301. ACM (2002)
14. Regan, K., Poupart, P., Cohen, R.: Bayesian reputation modeling in e-marketplaces sensitive to subjecthity, deception and change. In: Proceedings of the 21st National Conference on Artificial Intelligence, vol. 2, pp. 1206–1212. AAAI Press (2006)
15. Vose, D. (ed.): Risk Analysis: A Quantitative Guide, 3rd edn. Wiley (2008)
16. Dubois, D.: Possibility theory and statistical reasoning. Computational Statistics and Data Analysis 51, 47–69 (2006)
17. Ferson, S., Ginzburg, L.R.: Diferent methods are needed to propagate ignorance and variability. Reliability Engineering and System Safety 54, 133–144 (1996)
18. Masson, M.H., Denœux, T.: Inferring a possibility distribution from empirical data. Fuzzy Sets Syst. 157, 319–340 (2006)

19. Goodman, L.A.: On simultaneous confidence intervals for multinomial proportions. Technometrics 7, 247–254 (1965)
20. Yager, R.: On mean type aggregation. IEEE Transactions on Systems, Man, and Cybernetics 26, 209–221 (1996)
21. Zadeh, L.: Fuzzy sets. Information and Control 8, 338–353 (1965)
22. Dubois, D., Prade, H.: Fuzzy sets in approximate reasoning, part 1: inference with possibility distributions. Fuzzy Sets Syst. 40, 143–202 (1991)
23. Yager, R.R.: On the dempster-shafer framework and new combination rules. Inf. Sci. 41, 93–137 (1987)
24. Dubois, D., Prade, H.: When upper probabilities are possibility measures. Fuzzy Sets Syst. 49, 65–74 (1992)
25. Destercke, S.: Evaluating Trust from Past Assessments with Imprecise Probabilities: Comparing Two Approaches. In: Deshpande, A., Hunter, A. (eds.) SUM 2010. LNCS, vol. 6379, pp. 151–162. Springer, Heidelberg (2010)
26. Campos, L., Huete, J., Moral, S.: Probability intervals: a tool for uncertain reasoning. International Journal of Uncertainty, Fuzziness and Knowledge-based Systems 2, 167–196 (1994)
27. Dubois, D., Foulloy, L., Mauris, G., Prade, H.: Probability-possibility transformations, triangular fuzzy sets, and probabilistic inequalities. Reliable Computing 10, 273–297 (2004)

Building Self-adaptive Software Systems with Component, Services & Agents Technologies: *Self*-OSGi

Mauro Dragone

University College Dublin (UCD), Ireland
CLARITY Centre for Sensor Web Technologies, Belfield, Dublin, Ireland
mauro.dragone@ucd.ie
http://www.csi.ucd.ie/users/mauro-dragone

Abstract. This paper examines component & service, and agent technologies, and shows how to build a component & service-based framework with agent-like features for the construction of software systems with self-configuring, self-healing, self-optimizing, and self protecting (self-*) properties. This paper illustrates the design of one such framework, *Self*-OSGi, built over Java technology from the Open Service Gateway Initiative (OSGi) and loosely based on the Belief, Desire, Intention (BDI) agent model. The use of the new framework is illustrated and benchmarked with a simulated robotic application and with a dynamic service-selection test.

Keywords: Autonomic software, Self-* software systems, Agent oriented software engineering, Component based software engineering.

1 Introduction

Today, autonomic and adaptive software architectures are pursued in a number of research and application strands, including Robotics, cyber-physical systems, wireless sensor networks, and pervasive and ubiquitous computing.

In order to operate in these highly dynamic, unpredictable, distributed and open environments, these software systems must exhibit self-configuring, self-healing, self-optimizing, and self protecting (self-*) properties.

These problems are being addressed by both the Component-Based Software Engineering (CBSE) and the Agent-Oriented Software Engineering (AOSE) paradigms, each offering a modular design by which to encapsulate, integrate and organize the different systems functionalities

CBSE operates by posing clear boundaries between architectural modules (the *components*) and guiding the developers in re-using and assembling these components into applications. This typically involves an unambiguous description of the component's behavioral properties, and the set of their legitimate mutual relationships, in terms of provided and required interfaces (the *services*).

More recently, in order to adapt to varying resource availability and to increase system fault-tolerance, component frameworks are also provided with limited run-time flexibility through late-binding and dynamic wiring of component's interfaces.

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 300–316, 2013.

Noticeably, this makes component & service based systems already resemble AOSE solutions, by favouring a component-centric rather than a global system perspective. However, they fail to provide an adequate support for these adaptive implementations, in terms of a common adaptation model and/or adaptation steps. Thus, the developer has to write custom, application-specific adaptation code. This makes difficult to ensure that a consistent and interoperable adaptation strategy is applied throughout all the components in one application, and also to maintain, and re-use these strategies across multiple applications.

In contrast, AOSE provides a method of abstraction and system decomposition based on *agentification*. This transforms a software application into a goal-oriented, autonomous agent by building a wrapper around it so it can interoperate with the rest of the system through standard, Agent Communication Language (ACL) interfaces and associated coordination protocols.

However, the emphasis of AOSE and associated multiagent programming platforms and toolkits is the coordination of large scale, deliberative multiagent systems (MASs) while issues arising from low-level, application specific functionalities are often overlooked. Consequently, as noted in [8], using an agent platform limit the acceptance of mobile agents as simple programming constructs, as the programmer is forced to center its development, its programming units, and its whole applications on the concept of agent. Rather than a middleware-type complement to traditional (object/component-oriented) software development, agents then become the *frontware* and require the definition of complex interfaces toward the application and operating system resources.

These are the main motivations for seeking tightly integrated architectures that leverage the different characteristics and advantages of AOSE and CBSE. In particular, the focus of this work is the unification between agent, component and service concepts in a single methodology for the construction of autonomic software systems with Self-* properties. On one hand, this work aims to define a set of re-usable, modular end extensible adaptation mechanisms for component & service-based systems. On the other hand, the same approach will produce modular and lightweight agent systems that are tightly integrated with mainstream component & service technology.

The remainder of the paper is organized in the following manner: In order to draw a parallel between the CBSE and AOSE paradigms and guide the design of hybrid CBSE/AOSE systems, Section 2 summarises the popular Belief-Desire-Intention (BDI) agent model while Section 3 examines the most important characteristics of component & service frameworks. Section 4 draw upon the similarities between these technologies to translate the BDI agent model into component & service-based concepts. Section 5 introduces *Self*-OSGi, a novel agent toolkit, which is developed using the Open Service Gateway initiative (OSGi) component & service technology [12]. Section 6 illustrates the use of *Self*-OSGi and tests its performance with a robotic application and a dynamic service-selection example. Section 7 provides an overview of the most significant agent/component integration approaches attempted in past research. Finally, Section 8 summarizes the contributions of this paper and points to some of the directions to be explored in future research.

2 Agent Systems

The kernel of a particular agent-based architecture is usually based upon some abstract model of agency. This is executed on an agent platform and captured by an agent-programming language. A brief discussion of these key elements is essential to provide an overview of agent-based adaptive software solutions.

2.1 Agent Platforms

Although there is usually a degree of abstraction between language and architectural issues, a language will inevitably pose some constraints upon the underlying execution layer implementing the semantic of the language. The most commonly adopted approach in this regard is to structure the execution layer by distinguishing between agents and agent platforms, which then provide the functional bases upon which agents in a MAS can operate in their environment and interact with each other. In this way, an agent can be seen as an active software entity using the agent platform as a middleware to gain access to standardized services and infrastructure, such as life-cycle management, inter-agent communication, directory facilitators, coordination, security management, and mobility (migration).

Agent platforms do not only free the developer from low-level details but they also promote a basic level of modularity in the construction of the MAS as the platform services are re-used in each agent. By adhering to FIPA reference specifications, agent platforms such as JADE [1] and Agent Factory (AF) [2] also guarantee an important level of cross-platform interoperability. In addition, some platforms, such as Jade and AF, increase interoperability by not being tightly coupled to specific programming languages.

In systems such as Jade and Jack, the agent language is defined directly in terms of Java classes. These classes are extensions of a basic agent class and have direct access to the platform API. In AF, this pure-java option coexists with a number of interpreted languages, including AF-APL and AgentSpeak. These are defined by using a core library, which provides support for generic agent interpreters and for resolution based logic.

2.2 The BDI Agent Model

The Belief, Desire, Intention (BDI) is undoubtedly the most popular agent model, with many implementations directly related to Rao & Georgeff's abstract BDI architecture [13] and its Procedural Reasoning System (PRS) implementation [5].

Kinny et al. [4], describes the design of a BDI agent in terms of three components:

- A ***Belief Model***, describing the information about the environment and internal state that an agent may hold, together with the actions it may perform.
- A ***Goal Model***, describing the desires that an agent may possibly intend, and the events to which it can respond
- A ***Plan Model***, describing the set of plans available to the agent for the achievement of its goals

While these traits were introduced to allow the computational tractability of the model, they are now recognized as being the most distinctive characteristics of BDI systems in general for their ability to support rational, resource-bounded reasoning in dynamic and uncertain domains.

Beliefs, for example, are essential since an agent has limited sensory ability and needs to build up its knowledge of the world over time. In this sense, beliefs - usually represented with first order logic predicates - serve as a cache with which the agent can remember past events or other important information that could be costly to re-compute from raw perceptual data or inferred logically.

The distinctions between goals and plans constitute an important source of modularity that contributes to the agent's ability to cope with contingencies. The fundamental observation is that goals, as compared to plans, are more stable in any application domain and multiple plans can be used/attempted to achieve the same goals. This also allows examining the application domain in terms of what needs to be achieved, rather than the types of behaviour that will lead to achieving it.

PRS implements a computationally tractable BDI model with the following simplifying assumptions:

- The system explicitly represents beliefs about the current state of the world as a ground set of literals with no disjunctions or implications (as in STRIPS).
- The system represents the information about the means of achieving certain future world states and the options available to the agent as pre-compiled plans.

Each plan in the plan library can be described in the form of Event Condition Action (ECA) rules $e : \Psi \leftarrow P$ where P is the body of the plan, e is an event that triggers the plan (the plan's post-conditions), Ψ is the context for which the plan can be applied (which corresponds to the preconditions of the plan).

The body of each plan is a procedural description containing a particular sequence of actions and tests that may be performed to achieve the plan's post-condition. Plans are activated in response to the posting of new goals or upon reception of events. This process consists of finding the plans whose invocation unifies with the active goals or events and whose context unifies with the state contained in the agent's belief set. Plans may also post new goal events, leading to the characteristic AND/OR, goal/plan execution graphs. For a plan to succeed all the subgoals and actions of the plan must be successful (AND); for a subgoal to succeed one of the plans to achieve it must succeed (OR). When a plan step (an action or sub-goal) fails for some reason, this causes the plan to fail, and an alternative applicable plan for its parent goal is tried. If there is no alternative applicable plan, the parent goal fails, cascading the failure and search for alternative plans one level up the goal-plan tree.

In PRS-like BDI systems, desires and goals are represented only in the transient form of goal events (posted by the application), while the intentions to purse them is stored implicitly in the stack of plans they triggered. This poses an obstacle to the effective decoupling between plans and goals, forcing, for instance, the agent to drop goals for which no feasible plan can be attempted at the time the goal is posted. Such an issue is addressed in modern agent systems, such as Jadex, by incorporating explicit and declarative goal representations into the agent interpreter in order to ease the definition of goal deliberation strategies [3].

The ability to search for alternative applicable plans when a goal is first posted or when a previously attempted plan has failed is essential to enable these systems to handle dynamic environments. In some situations there can be multiple plan options to achieve a given goal, but for a given state, only certain combinations of choices will lead to its successful achievement. In PRS, as in modern BDI systems, the final deliberation of which plan to activate is usually performed using meta-level plans that operate upon meta-level descriptions of the other plans and the goals in the system. Meta-level plans are important hooks used by the designer to implement application-specific plan selection strategies, for example, by considering plan properties, e.g. reflecting quality of service, or to insure mutual exclusion on critical resources, and optimize their utilization.

3 Component and Services

The Open Service Gateway Initiative (OSGi), CORBA Component Model, Microsoft Object Model, Enterprise JavaBeans, and Fractal are some of the mainstream component-enabling technologies used for the creation of many industrial-strength software systems. Conceptually, the same technologies also provide a composite model for service oriented architectures, by helping to design systems in terms of application components that can expose their public functionality as services as well as invoke services from other components.

3.1 Component Containers

One of the most important common concepts among component-enabling technologies is the relationship between a component and its environment, wherein a newly instantiated component is provided with a reference to its container or component context. The component container can be thought of as a wrapper that deals with technical concerns such as synchronisation, persistence, transactions, security and load balancing. The component must provide a technical interface so that all components will have a uniform interface to access the infrastructure services. For instance, a common solution to implement activity-type components, i.e. components that need to attend to their process rather than merely react to events, is to segment these activities in steps, which are then executed by a scheduler - usually shared among multiple components.

Most relevant for this paper, OSGi defines a standardised component model and a lightweight container framework, built above the JVM. OSGi is used as a shared platform for network-provisioned components and services specified through Java interfaces. Each OSGi platform facilitates the dynamic installation and management of units of deployment, called bundles, by acting as a host environment whereby various applications can be executed and managed in a secured and modularised environment. An OSGi bundle organises the frameworks internal state and manages its core functionalities. These include both container and life cycle operations to install, start, stop and remove components as well as checking dependencies.

3.2 Component & Service Adaptation

Many of the infrastructural services associated with component contexts act as late-binding mechanisms that can be used to defer inter-component associations by locating suitable collaboration partners. Through these brokering mechanisms, components do not need to be statically bound at design/compilation time but can be bound either at composition-time or at run-time in order to favour the construction of adaptable software architectures.

For instance, the *Activator* class in OSGi, the *BeanContext*, and the components membrane in Fractal enable components to look up services in the frameworks service registry, register services, access other components, and install additional components within the local platform.

The separation between component's services and their actual implementation is the key to the creation of self-managing and adaptable architecture. In striving toward these solutions, a formal base is usually required to describe the provided and required features of individual components and also important semantic aspects, such as the correct way those features are to be used. With OSGi, developers can associate lists of name/value attributes to each service, and use the LDAP filter syntax for searching the services that match given search criteria. Furthermore, Declarative Services (DS) [6] for OSGi offers a declarative model for managing multiple components within each bundle and also for automatically publishing, finding and binding their required/provided services. This minimizes the amount of code a programmer has to write; it also allows service components to be loaded only when they are needed (Delayed Activation). Declarative Services indicates if a required service is mandatory or optional. The binding makes the life cycle of the component dependent on the presence of that linkage, respectively having its state as active or passive depending on the presence or absence of the component's dependencies.

These mechanisms are not limited to components and services running on a single platform. Remote service bindings are usually achieved through port and proxy mechanisms. For example, in both R-OSGi and D-OSGi, remote bindings can be viewed as connection points on the surface of the component where the framework can attach (connect) references to provides-ports provided by other components. The framework is then responsible for returning the correct Java object when a port is requested by a component. It either calls the appropriate methods of the locally available service implementation object or translates the Java method calls to messages, sends them to a remote container (e.g., availing of Java RMI, SOAP, or JXTA), waits for remote execution and then returns the value contained in the received message.

In the OSGi implementation OSCAR, the same mechanism is also used to support intelligent hot swapping of services to implement fault-tolerant systems. Specifically, as every service in OSGi may be given a certain rank which can be used to describe its quality and importance, when queried about a particular service, OSCAR automatically tries to locate the highest-ranked implementation.

A-OSGi [7] goes a step further by providing a number of mechanisms that can be used to create self-adaptive architectures. Firstly, a monitor component measures the CPU and memory used by each bundle by: (i) altering the OSGi life cycle layer so that all the threads in a bundle belong to the same thread group, and by (ii) providing each service client with a proxy that executes the service methods within the same thread group. Secondly, a planning component interprets ECA rules specifying adaptation actions (used to start, stop and configure bundles) to be executed in response to specific events and given conditions. Finally, an execution component applies these actions to their target components.

4 Component & Service Agent Model

The characteristic new approach advocated in this work is to address the lack of common adaptation mechanisms in component & service frameworks by leveraging their previously unexploited similarities with the BDI agent model.

The previous two sections allows us to draw many similarities between agent and component & service-based frameworks: while the first favours the construction of applications in terms of loosely coupled, autonomous entities (the agents), each agent is still a managed entity within an agent platform - just as components are managed within their container. Furthermore, similarly to the separation between goals and plans in agent systems, the decoupling between component's services and their actual implementation is the key to the creation of adaptive software systems.

In order to inform the design of a new generation of frameworks for adaptive software systems, and before dwelling on the details of its OSGi-based implementation (*Self*-OSGi), this section translates the BDI model into general component & service concepts.

4.1 Modular Belief Model

Rather than storing all the agent's beliefs into a single, centralized belief set, a component & service-based organization can be used to access and distribute the processing of information across the system. Specifically, an agent may use a number of sensor components to interface with its environment. Each of these components produces data that can be exported with any of the collaboration styles afforded by mainstream component & service-enabling technologies (procedural calls, messaging, events). This information can be fed to other perception components, for instance, to infer situations or test conditions involving multiple beliefs, thus enabling a variety of perception architectures to fit with the sensors available to the agent and also with the run-time requirements posed by its specific perceptual processes.

4.2 Service Goal Model

A ***Service Goal*** is informally defined as *the interface of a service that may be used to achieve one of the agent's goals*.

Service goals may represent either: (i) sub-goals defining the desired conditions to bring about in the world and/or in the system's state (for instance, the service *(void)* *atLocation(X, Y)* may be used by a robotic agent to represent the goal of being at a given location), and (ii) sub-goals subtending the exchange of information. Service goals that are used to access data and to subscribe to data updates and event notifications fall into the latter category. For instance, the service goal *Image getImageCamera()* may be used by a robotic agent to express the goal of retrieving the last frame captured by one of its cameras.

In addition, service goals' attributes may be used to further characterise each service goal, e.g. the characteristic of the information requested/granted, as well as important non-functional parameters. For instance the *atLocation* service goal may have the attribute *Min/MaxVelocity* to specify the minum/maximum velocity the robot should/may travel. The attribute *MinimumFrameRate* may be used to specify the mimum frame rate for the image captured with the *getImageCamera()* service, while the *Side* attribute, with values in $\{$*left, right*$\}$, may be used to specify which one of the robotic cameras must be used.

4.3 Component Plan Model

A ***Component Plan*** is informally defined as *a component implementing (providing) a service goal (its post-condition)*. A component plan may require a number of service goals in order to post sub-goals, to perform actions, and also to acquire the information it needs to achieve its post-condition. Component plans may attend their activities with their own thread of control. In addition, they may react to incoming messages/events, and also export functions to a scheduler used for control injection. For instance, a *MoveTo* component plan may process the images from a robot's cameras and control the velocity and the direction of the robot to drive it safely toward a given location. The same component plan may subscribe to impact alert notifications generated by the onboard bumper sensor, and stop the robot upon the reception of one such alert.

4.4 Goal Manager

Section 2 has discussed how an agent must rely on explicit representations of its own goals in order to keep track of goals achieved and yet to achieve.

To this end, it is useful to introduce the concept of ***Goal Manager***, that is, a component used to decouples the plan requesting a service goal from the component plan ultimately providing it. Invoking a service goal should first trigger the activation of the corresponding Goal Manager, which then will take care to invoke one of the component plans able to achieve it.

Thanks to its mediation, a Goal Manager can be used to re-invoke the same service goal upon failure of the component plan first used to achieve it. Crucially, further invocations may use different implementations of the service, i.e. different component plan options. In addition, the Goal Manager can be used to maintain execution statistics for each component plan option, in order to drive the future selection of the best suitable one, e.g. the one less likely to fail, and/or with better performance.

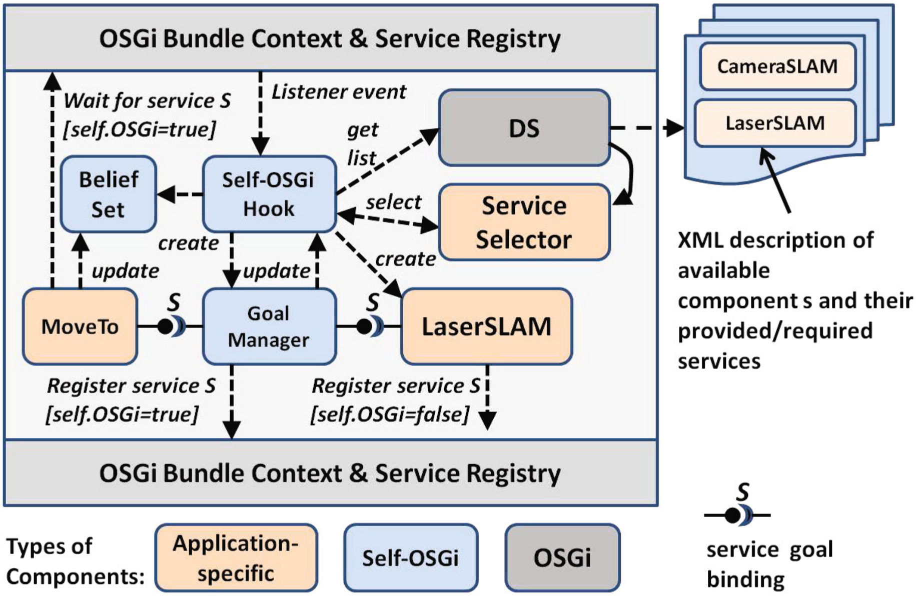


Fig. 1. *Self*-OSGi System Diagram

5 *Self*-OSGi Implementation

Figure 1 shows a diagram of the main classes involved in the operation of *Self*-OSGi, including pre-existing OSGi classes, and few application components (*MoveTo*, *LaserSLAM* and *CameraSLAM*), extracted from a robotic example application.

With its declarative execution model, components are activated and deactivated under the full control of the ***OSGi Declarative Service (DS)*** bundle. DS bases its decisions on the information in the components definition stored in XML files, which describe the services that are implemented by the component, and its dependencies on other services. DS will automatically register every service provided by the component into the central service registry of the platform, and bind every service required by the components with those available in the registry.

If a components description specifies the *factory* attribute of the component element in its XML definition, DS will register a ***OSGi Component Factory*** service. This service allows client bundles to create and activate multiple components' instances, on demand, and dispose of them after they have been used.

Self-OSGi provides an explicit resolution service, which can be used explicitly by application component plans to get the reference to the implementation of the service goals they need. In the example, the service can be used to get a reference to an implementation of the service goal *S* - representing any robot's localization system in the example - to be called by the component plan *MoveTo*. Alternatively, the developer can register the ***OSGi Framework Find Hook Service*** in order to be called during framework service find (*getServiceReferences()*) operations. Noticeably, the latter approach does not require any intervention to the application code, (which can be developed in terms of plain old java objects (POJOs), thus making very easy to apply *Self*-OSGi to pre-existing applications.

On its own, the OSGi DS is only able to satisfy a service request by looking at components already active or pending activation. However, the agent execution model requires the on-demand activation of component plans. For this reason,*Self*-OSGi collaborates with the ***OSGI Factory Service*** (not depicted in the diagram) and with the DS to create and bind component plans on demand.

Once called by the framework or called explicitly by component plans in need for services, *Self*-OSGi creates a ***Goal Manager***, which will install a ***GoalProxy*** object (implemented using the Java *dynamic proxy* class) by registering it to the OSGi service registry in place of the original requested service.

After that, the DS will automatically bind the *GoalProxy* to the component plan that has requested the service goal. However, only when the service goal is finally invoked, the *GoalManager* will trigger the selection of a suitable service goal's implementation. To do this, *Self*-OSGi will (i) find all the components suitable to be executed in the current context (options), and (ii) select and instantiate the one that is deemed to be the best to satisfy the request at hand.

In order to carry out the first step, *Self*-OSGi enriches the OSGi container and the DS' OSGi XML component descriptions to drive its agent-like, dynamic management of component plans' dependencies and instantiation. First of all, each component plan can commit part of its internal variables into a *Self*-OSGi *Belief Set* component. Such components export a simple get/query/set/delete API that can be used to query the belief set for specific data or for beliefs matching a LDAP filter; and to store and remove the resulting data from the belief set. Secondly, *Self*-OSGi introduces variable-type properties that can be used to refer to the beliefs stored in these belief sets, and also to specify dependencies between the properties of the post-condition of component plans (their provided service goal) and the ones of their sub-goals (their required service goals). *Self*-OSGi is able to use LDAP queries, including logic and mathematical relations, that are defined over the properties currently held in the agent's belief set. In this manner, *Self*-OSGi can quickly skip all the unsuitable options before asking the *Self*-OSGi's *ServiceSelector* to rank them, and finally instantiate the first option in the ranking order (*LaserSLAM* in the example).

The *GoalProxy* installed with the *GoalManager* measures execution statistics (CPU system and user time) for both synchronous and asynchronous invocation to its corresponding service goal. A number of dedicated attributes in the original service goal request can be used to alter the call semantic, including *MaxAttempts* (used to specify the number of invocation to be attempted upon failure), and *DelayBetweenAttempts* (used to specify a delay between each attempts). Crucially, after each failed attempt, the goal proxy will contact the *Goal Manager*, in order to trigger the resolution of a (possibly) different implementation (plan option).

The *Self*-OSGI architecture includes other components that are not displayed in the diagram of Fig. 1. The ***Self*-OSGI *GoalHandler*** class may be used to support asynchronous operations. If the requester registers a *GoalHandler*, the goal proxy will schedule each invocation with a platform's scheduler (implemented over the Java

ExecutorService class). Group-type specializations of the goal handler class exist, such as *ANDGoalHandler* and *ORGoalHandler*, which are used, respectively, to define conjunctions and disjunctions of groups of service goals. Noticeably, using group goal handlers breaks the POJO programming model, as the developer needs to incorporate these *Self*-OSGi specific classes into its code.

Finally, the ***Self*-OSGI *PerformanceMonitor*** class collects the performance statistics for all the component plan options. This class associates a priority to each component option. By default, the priority is computed by considering both the success rate (the rate between the number of successful invocation and the total number of invocation) and the speed to which the component had satisfied past requests, but the developer can easily fit other, application or domain-specific ranking mechanisms. As other features of *Self*-OSGi, this is done by leveraging the OSGi service registry to register ranking components, execution schedulers, and other application or domain-specific service, which will be then retrieved and used by *Self*-OSGi.

6 Tests

Self-OSGi is being used to refactor a number of pre-existing applications, including robot control and distributed information retrieval systems. Developers can add Self-* capabilities to their OSGi applications by using the Self-OSGi Core bundle (which consumes 31K on the file system).

6.1 Robotic Example

As a way of example, the following is part of the XML documents describing a *MoveTo*, a *CameraLocalization* and a *LaserLocalization* component plans used by a mobile robot agent.

In order to clarify its correspondence with the BDI model it represents, each XML is preceded with a comment in the ECA form $e : \Psi \leftarrow P$ where P is the body of the plan, e is the event that triggers the plan (the plan's post-conditions), and Ψ is the context for which the plan can be applied (which corresponds to the preconditions of the plan).

$GoalMoveTo(?Agent) : true \leftarrow \{achieve(GoalLocalization(?Agent)); ...\}$

```
<scr:component ... factory="Navigation" name="MoveTo">
<implementation class= "MoveTo"/>
<property name="?Agent" type="String" value="The name of the robot
                                                   supposed to move">
<service>
          <provide interface="GoalNavigation"/>
</service>
<reference cardinality="0..1" interface= "GoalLocalization"
                policy="dynamic" target="(Agent=?Agent)>
</scr:component>
```

$GoalLocalization(?Agent) : (light > 30) \leftarrow \{achieve(GoalVideo(?Agent)); ...\}$

```
<scr:component ... factory="CameraLocalization"
                     name="CameraLocalization">
```

```
<implementation class= "CameraLocalization"/>
<property name="?Agent" type="String" value="The name of the robot to
                                                        be localized">
<service>
         <provide interface="GoalLocalization"/>
</service>
<reference cardinality="0..1" interface= "GoalVideo" name="Video"
                                          target="(Agent=?Agent)/>
<property name="self.osgi.precondition.LDAP" value="(light>30)"/>
</scr:component>
```

$GoalLocalization(?Agent) : true \leftarrow \{achieve(GoalLaser(?Agent)); ...\}$

```
<scr:component ... factory="LaserLocalization"
                    name="CameraLocalization">
<implementation class= "LaserLocalization"/>
<property name="?Agent" type="String" value="The name of the robot to
                                                        be localized">
<service>
          <provide interface="GoalLocalization"/>
</service>
<reference cardinality="0..1" interface= "GoalLaser" name="Laser"
                                          target="(Agent=?Agent)/>
</scr:component>
```

Post-conditions. The post-conditions of both component plans are specified with the OSGi XML *service* element. The *MoveTo* component plan implements a move-to navigation behaviour in order to provide the service goal *GoalNavigation*, while both the *CameraLocalization* and the *LaserLocalization* component plans implement localization methods in order to provide localization updates through the service goal *GoalLocalization*.

Service Goal Requirements. Service goal requirements are declared using OSGi XML *reference* elements. The definition of *MoveTo* declares its requirement of localization information as *dynamic*, in order to allow OSGi to activate it even when the reference to the *Localization* service goal is not resolved, thus avoiding to having to commit to a specific localization mechanism before the behaviour is started.

Noticeably, the definition of *CameraLocalization* includes *Self*-OSGi-specific property fields, *self.osgi.precondition.LDAP*, whose value may be used to characterise the context when the component plan is applicable. In the example, the LDAP pre-condition describes how *CameraLocalization* can only be used when the intensity of the ambient light, e.g. sensed by a light sensor component, is believed to be above a given threshold.

Variables. In order to link post-conditions with pre-conditions and service goal requirements, *Self*-OSGi allows the use of variable attributes whose name starts with the special character ”?”. Variables may be used as names of the property associated to a component plan in order to specify that the component plan can be instantiated with any value for that property. In such a case, the value of the variable is used as default value of the property. For instance, both the *MoveTo* and the *CameraLocalization* component plans declare the property *Agent* to specify that they can be used by any agent to achieve, respectively, the *GoalNavigation* and the *GoalLocalization* service goals. Once the respective component plans have been instantiated for a specific robot agent, i.e. *Robot1*, the services they provide will have an *Agent* attribute with value *Robot1*. However, in order for the same services to work, they must receive updates, respectively, of location and video data related to the same robot agent. Both XML descriptions specify this dependency by repeating the attribute *?Agent* in the reference elements describing their required service goals. *Self*-OSGi will take care to propagate the value of these variable properties from the post-condition/service side to the requirement/reference side, for instance, to wire a *MoveTo* component activated for a *Robot1* robot agent, with a *LaserLocalization* activated for the same robot agent, rather than using the pure syntactic match (which could be satisfied by any localization data, e.g. related to other robots or used to represent the location of a human user).

Finally, the following code is used to send a robot to a given location by initializing a standard OSGi *ServiceTracker* object to request the *GoalNavigation* service goal, before invoking it by passing the location coordinates. The special attribute *selfosgi=true* is used to demand the *Self*-OSGi management of the call. Noticeably, no other modifications are required to standard OSGi programming.

```
ServiceTracker tracker = new ServiceTracker(...,
context.createFilter("(\&(objectClass
    ="+GoalNavigation.class.getName()+") (selfosgi=true)").open();
(GoalNavigation)(tracker.waitForService(0)).beAt (100, 200);
```

In the localization example, these features have been used to make the robot reach its destination while opportunistically exploiting any suitable localization mechanism, for instance, starting with the *CameraLocalization* and then switching to the *LaserLocalization* if the first fails, e.g. when the ambient light drops below the given threshold.

In order to measure the time needed by *Self*-OSGi to perform the reconfiguration, tests were performed using a non-physically realistic simulated robot system, whose hardware components had no initialization latency, and no inertia. Over 10 runs using a Pentium dual core at 2.40 GHz with 8GB of SDRAM, with the Sun J2SE 6.0 platform compliant JVM and running the Linux 2.6.24 kernel, the system was able to recover from failure with an average of 72ms with standard deviation of 3.78. For comparison, a similar dynamic service replacement performed over the Java Beans component framework and co-ordinated by a standard BDI agent toolkit took an average of 312ms [11].

The *Self*-OSGi's management of real and more complex robotic systems has been recently enabled by integrating *Self*-OSGi with the PEIS (Ecologies of Physically Embedded Intelligent Systems) middleware [16], and by making it operate across distributed platforms, thanks to the R-OSGi distributed extension of OSGi [15].

6.2 Service-Selection

Self-OSGi includes a *Benchmark* bundle to enable developers to assess *Self*-OSGi performance and estimate the overhead it imposes to specific applications. To this end, the Benchmark bundle implements a simple simulated test domain, which models a typical *Self*-OSGi application where both the set of available component plans, and the set of service goals are organized in *L* levels of abstraction.

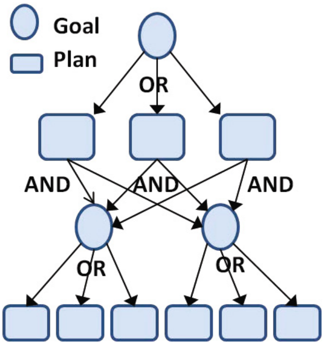


Fig. 2. Structure of the test program

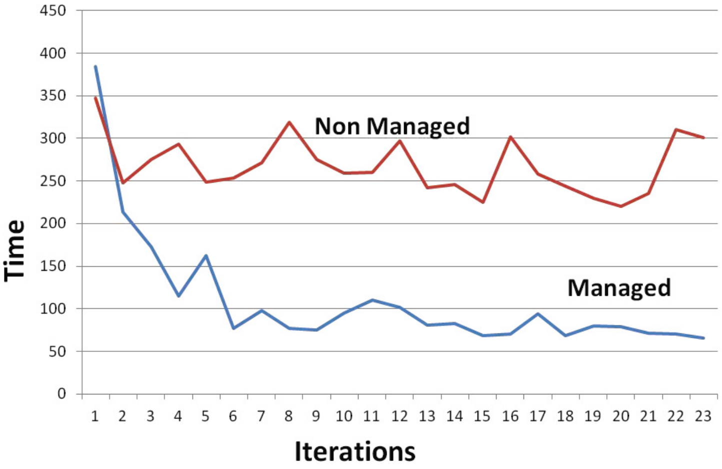


Fig. 3. Service selection performance

Figure 3 shows the execution times obtained with $L = 2$ by using component plans requiring different CPU times for the execution of the service goals they provide, with and without installing the *Self*-OSGi's GoalManager - rispectively, the managed and not-managed cases depicted in Figure 3. In the former case, a service ranking component assigning greater priorities to previously unexplored plan options was used. The figure demonstrates how, when *Self*-OSGi was repeatedly asked to achieve the root goal, it automatically tried new component plan implementations at each iteration, ultimately converging on the best policy to achieve the root goal in the shortest time.

With the same setup used in the robotic example, the overhead imposed by the *Self*-OSGi service management (performance measurement and proxy mechanisms) was measured by comparing the time needed to achieve the root service goal with and without the installation of a GoalProxy. The average time added by the proxy service management to each service goal was 0.15 milliseconds.

7 Related Work

A number of works have looked at leveraging both component and agent approaches for the development of adaptive software systems.

A component-based approach in the construction of multi agent systems has been supported by numerous researchers in the past. This typically considers the components to be simply the building blocks from which agents are constructed [10]. An advantage of this approach is the ability to take domain-specific issues into account at the component level. The decisions made on these issues can therefore be separated from the task of constructing the multi agent system as a whole, thus simplifying the process. The resulting agent applications inherit some of the (functional/non-functional) properties from the underlying component framework. However, this form of technical integration does not contribute much to a conceptual combination of both paradigms as, once they are built, agents remain the only primary entity form.

A different integration approach is advocated in SoSAA [11], in which an high-level agent framework supervises a low-level component-based framework. The latter provides a computational environment to the first, which then augments its capabilities with its multi-agent organisation, ACL communication, and goal-oriented, BDI-style reasoning. A SoSAA Adapter interface provides meta-level sensors and meta-level actuators to operate on the component layer, to load, unload, configure components, observe their internal status, and bind their provided/required interfaces. Components are left to automatically carry out lower-level behaviours and can interact through a variety of non-ACL collaboration styles, including method calls, messages and events. The deliberative layer makes decisions about when such behaviours and communication mechanisms are necessary or desirable in order to satisfy overall system and application requirements. However, keeping neatly separated components and agents fails to contribute much in consolidating both paradigms. Furthermore, the use of two separate frameworks means that the resulting systems are subjected to both development and run-time overheads.

Removing the need for a separate infrastructure shared by a large number of distributed applications is what motivates the approach followed in the M&M framework [8]. In contrast to application development centred upon agent platforms, M&M applications become agent-enabled by incorporating well-defined binary software components into their code. These components give the applications the capability of sending, receiving and interacting with mobile agents. The applications themselves are developed using the current industry best practice software methods (*JavaBeans*) and become agent-enabled by integrating the mobility components. Such an approach succeeds in moving some agent mechanisms into the middleware layer. However, M&M only addresses agent mobility while components are not equipped with goal-oriented reasoning capabilities.

More recently, the *Active Component* (AC) concept [9] has been proposed as a way to integrate successful concepts from agents and components as well as active objects and make those available under a common umbrella. Active components are autonomous acting entities (like agents) that can use message passing as well as method calls (like active objects) for interaction. They may be hierarchically structured and are managed by an infrastructure that ensures important non-functional properties (like components).

The AC framework has been realized in the Jadex AC platform. In particular, Jadex AC runs on an extended Jade platform and supports component types (kernels) for BDI agents, as well as simpler, task-specific agent models. Noticeably, such an approach defines a proprietary component or agent frameworks and does not leverage mainstream component-based initiatives and standards, such as OSGi.

8 Conclusions and Future Work

This paper has examined component, service and agent concepts, and has illustrated the design and the implementation of the *Self*-OSGi framework for the construction of systems with Self-* properties. *Self*-OSGi is built over OSGi technology by leveraging previously unexploited similarities between component & service and the BDI agent model.

Compared to similar CBSE initatives, such as the A-OSGi framework reviewed in Section 3.2, *Self*-OSGi provides re-usable, lightweight, modular end extensible adaptation mechanisms at component-level granularity that are also tightly integrated with the OSGi Declarative Service framework. *Self*-OSGi can be used to drive the selection of services, control the on-demand instantiation of the components implementing them, and monitor their performance to drive their future selection and to recover from failure. In contrast, A-OSGi can be used to control and monitor entire bundles, but does not offer any mechanism to discern the performance among the single components and services inside the bundle or to instantiate them on-demand.

In addition, the association with the BDI model allows *Self*-OSGi to leverage well-defined adaptation policies and results from BDI-related research.

Compared to existing AOSE/CBSE integration approaches, such as the SoSAA and the AC framework reviewed in Section 7, *Self*-OSGi provides a highly modular realization of the BDI agent model, which is grounded in the mechanisms offered by a mainstream component & service technology. This results in low performance and footprint overheads and fast system's adaptation, as shown in Section 6. Noticeably, existing agent platforms, such as JADE, have already been made compatible with the OSGi framework. However, this is usually done by encapsulating the entire agent platform into a single, monolithic OSGi bundle. Such an approach does not benefit of the increased modularity enabled by the OSGi framework.

In contrast, one of the goal of the *Self*-OSGi framework is to evolve into a modular and interoperable agent platform. In addition, future research work with *Self*-OSGi will seek to adapt agent/planning integration and agent learning techniques to tackle some of the main limitations of adaptive component & service frameworks (which are common to basic procedural agent systems), such as their lack of look-ahead capabilities and their reliance on hard-coded pre-conditions of component plans.

Acknowledgements. This work has been supported by the EU FP7 RUBICON project (contract n. 269914).

References

1. Bellifemine, F., Poggi, A., Rimassa, G.: JADE - A FIPA compliant agent framework. In: Proc. of 4th International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents (1999)
2. Collier, R.W.: Agent Factory: A Framework for the Engineering of Agent-Oriented Applications. Ph.D. dissertation, Dept. of Computer Science, University College Dublin (1997, 2001)
3. Pokahr, A., Braubach, L., Lamersdorf, W.: A Goal Deliberation Strategy for BDI Agent Systems. In: Eymann, T., Klügl, F., Lamersdorf, W., Klusch, M., Huhns, M.N. (eds.) MATES 2005. LNCS (LNAI), vol. 3550, pp. 82–93. Springer, Heidelberg (2005)
4. Kinny, D., Georgeff, M., Rao, A.: A Methodology and Modeling Technique for Systems of BDI Agents. In: Perram, J., Van de Velde, W. (eds.) MAAMAW 1996. LNCS, vol. 1038, pp. 56–71. Springer, Heidelberg (1996)
5. Georgeff, M., Lansky, A.: A system for reasoning in dynamic domains: Fault diagnosis on the space shuttle. Technical Note 375, Artificial Intelligence Center, SRI International (1985)
6. Cervantes, H., Hall, R.S.: Automating service dependency management in a service-oriented component model. In: Proceedings of the Sixth Component-Based Software Engineering Workshop (2003)
7. Ferreira, J., Leitao, J., Rodrigues, L.: A-OSGi: A framework to support the construction of autonomic osgi-based applications. Technical Report RT/33/2009 (May 2009)
8. Marquest, P., et al.: A component-based approach for integrating mobile agents into the existing web infrastructure. In: 2002 Symposium on Applications and the Internet, SAINT 2002 (2002)
9. Braubach, L., Pokahr, A.: Addressing Challenges of Distributed Systems Using Active Components. In: Brazier, F.M.T., Nieuwenhuis, K., Pavlin, G., Warnier, M., Badica, C. (eds.) Intelligent Distributed Computing V. SCI, vol. 382, pp. 141–151. Springer, Heidelberg (2011)
10. Amor, M., Fuentes, L., Troya, J.M.: Putting Together Web Services and Compositional Software Agents. In: Cueva Lovelle, J.M., Rodríguez, B.M.G., Gayo, J.E.L., del Ruiz, M.P.P., Aguilar, L.J. (eds.) ICWE 2003. LNCS, vol. 2722, pp. 44–53. Springer, Heidelberg (2003)
11. Dragone, M., Collier, R.W., Lillis, D., O'Hare, G.M.P.: Practical Development of Hybrid Intelligent Agent Systems with SoSAA. In: Coyle, L., Freyne, J. (eds.) AICS 2009. LNCS, vol. 6206, pp. 51–60. Springer, Heidelberg (2010)
12. OSGi. Open Service Gateway Initiative (OSGi) (2011), `http://www.osgi.org/Main/~HomePage` (accessed October 2011)
13. Rao, A., Georgeff, M.: An abstract architecture for rational agents. In: Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference KR (1992)
14. Behrens, T.M., Dix, J., Hindriks, K.V.: Towards an environment interface standard for agent-oriented programming (a proposal for an interface implementation). Technical report, Clausthal University (2009)
15. Dragone, M., Swords, D., Abdel Naby, S., Broxvall, M.: A Programming Framework for Multi Agent Coordination of Robotic Ecologies. In: Tenth International Workshop on Programming Multi-Agent Systems, ProMAS (2012)
16. Saffiotti, A., Broxvall, M., Gritti, M., LeBlanc, K., Lundh, R., Rashid, J., Seo, B.S., Cho, Y.J.: The PEIS-Ecology Project: vision and results. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Nice, France (September 2008)

Part II
Agents

Asset Value Game and Its Extension: Taking Past Actions into Consideration

Jun Kiniwa[1], Takeshi Koide[2], and Hiroaki Sandoh[3]

[1] Department of Applied Economics, University of Hyogo, Kobe, Japan
[2] Department of Intelligence and Informatics, Konan University, Kobe, Japan
[3] Graduate School of Economics, Osaka University, Toyonaka, Osaka, Japan
kiniwa@econ.u-hyogo.ac.jp, koide@konan-u.ac.jp,
sandoh@econ.osaka-u.ac.jp

Abstract. In 1997, a minority game (MG) was proposed as a non-cooperative iterated game with an odd population of agents who make bids whether to buy or sell. Since then, many variants of the MG have been proposed. However, the common disadvantage in their characteristics is to ignore the past actions beyond a constant memory. So it is difficult to simulate actual payoffs of agents if the past price behavior has a significant influence on the current decision. In this paper we present a new variant of the MG, called an asset value game (AG), and its extension, called an extended asset value game (ExAG). In the AG, since every agent aims to decrease the mean acquisition cost of his asset, he automatically takes the past actions into consideration. The AG, however, is too simple to reproduce the complete market dynamics, that is, there may be some time lag between the price and his action. So we further consider the ExAG by using probabilistic actions, and compare them by simulation.

Keywords: Multiagent, Minority game, Mean asset value, Asset value game, Contrarian, Trend-follower.

1 Introduction

Background. A minority game (MG) has been extensively studied since it was originally proposed [7]. It is considered as a model for financial markets or other applications in physics. It is a non-cooperative iterated game with an odd population size N of agents who make bids whether to buy or sell. Since each agent aims to choose the group of minority population, he is called a *contrarian*. Every agent makes a decision at each step based on the prediction of a strategy according to the sequence of the m most recent outcomes of winners, where m is said to be the memory size of the agents. Though MG is a very simple model, it captures some of the complex macroscopic behavior of the markets.

It is also known that the MG cannot capture large price drifts such as bubble/crash phenomena, but just can do the stationary state of the markets. This can be intuitively explained as follows. Suppose that a group of buyers can keep a majority for a long time. Then a group of sellers must continuously win in the bubble phenomenon. However, since every agent wants to win and thus joins the group of sellers one after another,

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 319–331, 2013.

it will gain a majority soon. That is, the group of buyers cannot keep a majority, a contradiction. Thus, it is difficult to simulate the bubble phenomenon by MG.

Related Work. Much work has been done for the purpose of adapting MG to a real financial market. For example, first, several authors investigated the majority game (MJ), consisting of *trend-followers*. Marsili [14] and Martino et al. [15] investigated a mixed majority-minority game by varying the fraction of trend-followers. Tedeschi et al. [17] considered agents who change themselves from contrarians to trend-followers, and vice versa, according to the price movements. Second, another way is to incorporate more realistic mechanism. A grand canonical minority game (GCMG) [5,10,11] is considered as one of the most successful models of a financial market. In the GCMG, a set of agents consist of two groups, called producers and speculators, and the speculators are allowed not to trade in addition to buy and sell. Third, it is also useful to improve the payoff function. Andersen and Sornette[1] proposed a different market payoff, called $-game, in which the timing of strategy evaluation is taken into consideration. Ferreira and Marsili[9] compared the behavior of the $-game with that of the MG/MJ. The difficulty of the $-game is to evaluate its payoff function because we have to know one step future result. Kiniwa et al. [12] proposed an improved $-game, in which the timing of evaluation is delayed until the future result is turned out. Fourth, there are some other kinds of improvement. Liu et al. [13] proposed a modified MG, where agents accumulate scores for their strategies from the recent several steps. Recent work by Challet [4] proposed a more sophisticated model using asynchronous holding positions which are driven by some patterns. Finally, two books [6,8] comprehensively described the history of minority games, mathematical analysis, and their variations. Beyond the framework of MG, efforts to reproduce the real market dynamics are continued [16,18].

Motivation. The purpose of this paper is also to improve MG by the thirdly mentioned above. Though the framework of MG and its variants seem to be reasonable, we have a basic question — "Do people always make decisions by using their strategies depending on the recent history ?" Some people may just take actions by considering losses and gains. For example, if one has a company's stock which has rapidly risen (resp. fallen), he will sell (resp. not sell) it soon without using his strategy as illustrated in Figure 1. Such a situation gives us the idea of an acquisition cost, or a mean asset value. In the conventional games, like the original MG, an agent forgets the past events and makes a decision by observing only the price up/down within the memory size [1]. In our game, however, each agent evaluates the strategies by whether or not the current price exceeds his mean asset value. Since the mean asset value contains all the past events in a sense, he can increase his net profit by reducing the mean asset value. We call the game an *asset value game*, denoted by AG.

However, there is still an unsolved problem in AG that stems from the framework of MG: the payoff function does not give an action, but just adds points to desirable strategies. Thus, if the adopted strategy is not desirable, the agent has to wait until the

[1] Recently, several studies [2,3] in this direction have been made from the viewpoint of evolutionary learning.

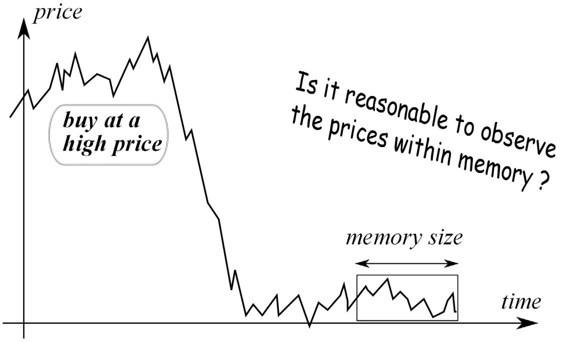


Fig. 1. Illustration of our idea

desirable one gains the highest score. So, there is a time lag between the rapid change of a price and the adjustment of an agent's behavior.

To improve the time lag, we allow each agent another action that satisfies the payoff function with some probability. If the price rises/falls rapidly and the difference between the price and agent i's mean asset value exceeds some threshold, the agent i may take the action according to the payoff function (regardless of the strategy). By tuning up the threshold, etc., we can reproduce the real market dynamics. We call the variant of AG an *extended asset value game*, denoted by ExAG.

Contributions. Our contributions in this paper are summarized as follows:

- We present a new variant of the MG, called an asset value game.
- To improve the problem of AG, we further consider an extended AG.
- We investigate the behavior of AG and ExAG in detail.

The rest of this paper is organized as follows. Section 2 states our model, which contains MG, MJ, AG and ExAG. Section 3 presents an analysis of AG. Section 4 describes a simulation model and shows some experimental results. Finally, Section 5 concludes the paper.

2 Models

In this section, we first describe MG and MJ in Section 2.1, then the difference between MG and AG in Section 2.2. Finally, we describe the difference between AG and ExAG in Section 2.3.

2.1 Previous Model — MG and MJ

At the beginning of the game, each agent $i \in \{1, \ldots, N\}$ is randomly given s *strategies* $\mathbf{R}^{i,a}$ for $a \in \{1, \ldots, s\}$. The number of agents, N, is assumed to be odd in order to break a tie. Any strategy $R_{i,a}(\mu) \in \mathbf{R}^{i,a}$ maps an m-length binary string μ into a decision -1 or 1, that is,

$$\mathbf{R}^{i,a} : \{-1, 1\}^m \longrightarrow \{-1, 1\}, \tag{1}$$

where m is the memory of agents. A *history* H, e.g., $[-1, 1, 1, \ldots]$, is a sequence of -1 and 1 representing a *winning decision* $h(t)$ for each time step $t \in T = (0, 1, 2, \ldots)$. The winning decision of MG (resp. MJ) is determined by the minority (resp. majority) group of -1 or 1. Each strategy $R_{i,a}(\mu) \in \mathbf{R}^{i,a}$ is given a *score* $U_{i,a}(t)$ so that the best strategy can make a winning decision. For the last m winning decisions, denoted by $\mu = h^m(t-1) \subseteq H$, agent i's strategy $R_{i,a}(\mu) \in \mathbf{R}^{i,a}$ determines -1 or 1 by (1). Among them, each agent i selects his highest scored strategy $R_i^*(\mu) \in \mathbf{R}^{i,a}$ and makes a decision $a_i(t) = R_i^*(\mu)$ at time $t \in T$. The highest scored strategy is represented by

$$R_i^*(\mu) = \arg \max_{a \in \{1,\ldots,s\}} U_{i,a}(t), \tag{2}$$

which is randomly selected if there are many ones. An aggregate value $A(t) = \sum_{i=1}^{N} a_i(t)$ is called an excess demand. If $A(t) > 0$, agents with $a_i(t) = -1$ win, and otherwise, agents with $a_i(t) = 1$ win in MG, and vice versa in MJ. Hence the payoffs g_i^{MG} and g_i^{MJ} of agent i are represented by

$$g_i^{MG}(t+1) = -a_i(t)A(t) \text{ and} \tag{3}$$

$$g_i^{MJ}(t+1) = a_i(t)A(t), \text{ respectively.} \tag{4}$$

The winning decision $h(t) = -1$ or 1 is added to the end of the history H, i.e., $h^{m+1}(t) = [h^m(t-1), h(t)]$, and then it will be reflected in the next step. After the winning decision has been turned out, every score is updated by

$$U_{i,a}(t+1) = U_{i,a}(t) \oplus R_{i,a}(\mu) \cdot sgn(A(t)), \tag{5}$$

where $\oplus$ means subtraction for MG (addition for MJ) and $sgn(x) = 1\ (x \geq 0),\ = -1\ (x < 0)$. In other words, the scores of winning strategies are increased by 1, while those of losing strategies are decreased by 1. We simply say that an agent increases selling (resp. buying) strategies if the scores of selling (resp. buying) strategies are increased by 1. Likewise the decrement of scores. Notice that the score is an accumulated value from an initial state in the original MG. In contrast, we define it as a value from the last H_p steps according to [13]. That is, we use

$$U_{i,a}(t+1) = U_{i,a}(t) \oplus R_{i,a}(\mu) \cdot sgn(A(t)) - U_{i,a}(t - H_p). \tag{6}$$

The constant H_p is not relevant to m, but is only used for selecting the highest score. Analogous to a financial market, the decision $a_i(t) = 1$ (respectively, -1) represents buying (respectively, selling) an asset. Usually, the price of an asset is defined as

$$p(t+1) = p(t) \cdot \exp \frac{A(t)}{N}. \tag{7}$$

2.2 Asset Value Game

The difference between MG and our asset value game is the payoff function. Let $v_i(t)$ be agent i's mean asset value at time t, and $u_i(t)$ the number of units of his asset. The payoff function in AG is defined as

$$g_i^{AG}(t+1) = -a_i(t)F_i(t), \tag{8}$$

where $F_i(t) = p(t) - v_i(t)$. The mean asset value $v_i(t)$ and the number of asset units $u_i(t)$ are updated by

$$v_i(t+1) = \frac{v_i(t)u_i(t) + p(t)a_i(t)}{u_i(t) + a_i(t)} \tag{9}$$

and

$$u_i(t+1) = u_i(t) + a_i(t), \tag{10}$$

respectively. That is, the payoff function (3) in MG is replaced by (8) in AG. Without loss of generality, we assume that $v_i(t), u_i(t) > 0$ for any $t \in T$.

The basic idea behind the payoff function is that each agent wants to decrease his acquisition cost in order to make his appraisal gain. Figure 2(a) shows the relationship between the price and the mean asset values of $N = 3$ agents, where the price is represented by the solid, heavy line. Notice that if the population size N is small, the price change becomes drastic.

The most important feature of the AG is to appreciate the past gains and losses. Even though an agent has bought a high-priced asset during the asset-inflated term (see Figure 1), the mean asset value of the agent reflects the fact and an appropriate action compared with the current price is recommended.

2.3 Extended Asset Value Game

Here we consider the drawbacks of AG, and present an extended AG, denoted by ExAG, to improve them. Though the AG captures a good feature of an agent's behavior, the payoff function indirectly appreciates desirable strategies. If the adopted strategy is not desirable, the agent has to wait until the desirable one gains the highest score. So, there is a time lag between the rapid change of a price and the adjustment of an agent's behavior.

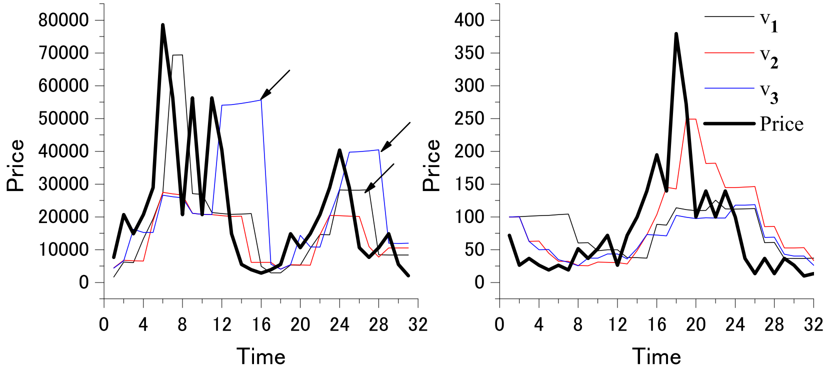
More precisely, the movement of price is followed by the asset values (see arrows in Figure 2(a)). This behavior can be explained by the following reasons. If the price rapidly rises, it exceeds almost all the mean asset values. Then, $F_i(t) = p(t) - v_i(t)$ becomes plus and the $a_i(t) = -1$ (i.e., sell) action is recommended. So, some agents change from trend-followers to contrarians in a few steps. During the steps, such agents remain trend-followers, that is, buy assets at the high price. Thus, their mean asset values follow the movement of price.

trim = 10mm 80mm 20mm 5mm, clip, width=3cm

Our solution is to provide another option of the agent. That is, the agent who has much higher/lower asset value than the current price can directly act as the payoff function, called a *direct action*. However, if so, every agent may take the same action when the price go beyond every asset value. To avoid such an extreme situation, we give the direct action with some probability.

Let $K = K^+ \ (F_i \geq 0),\ K^- \ (F_i < 0)$ be the F_i's threshold over which the agent may take the direct action, and let λ be some constant. Each agent takes the same action as the payoff function (without using his strategy) with probability

$$p = \begin{cases} 1 - \exp\{-\lambda(|F_i| - K)\} & (K \leq |F_i|) \\ 0 & (|F_i| < K), \end{cases} \tag{11}$$



(a) Asset value game (b) Extended asset value game

Fig. 2. Price influence on mean asset values

where

$$K = \begin{cases} K^{+} & 0 \leq F_i \\ K^{-} & F_i < 0 \end{cases} \quad \text{such that } K^{-} < K^{+}$$

and takes the action according to his strategy with probability $1 - p$. In short, in ExAG

- agent i takes an action $a_i(t)$ satisfying $g_i^{AG}(t+1) > 0$ with probability p, and
- an action $a_i(t) = R_i^*(\mu)$ with probability $1 - p$.

Figure 2(b) shows the behavior of the price and the mean asset values for $N = 3$ agents in our extended AG, where $K^{+} = 300$, $K^{-} = 50$ and $\lambda = 0.001$. Notice that the change of price in Figure 2(b) is not so drastic as that in Figure 2(a). In addition, all the values do not follow the price movement.

3 Analysis of AG

In this section we briefly investigate the features of AG. Though we mainly discuss the bubble in the following, similar arguments hold for the crash. For convenience, we define a contrarian as follows. If $a_i(t) = -1$ (resp. $a_i(t) = 1$) for a history $h^m(t-1) = \{1\}^m$ (resp. $h^m(t-1) = \{-1\}^m$), agent i is a contrarian. Let t_r be the first time at which the winning decision is reversed after $t - m$. Let $C^{MJ}(t)$, $C^{AG}(t)$ and $C^{MG}(t)$ denote the set of contrarians in MJ, AG and MG, respectively. The next theorem means that the bubble phenomenon is likely to occur in the order of MJ, AG and MG.

Theorem 1. *Suppose that the same set of agents experience $h^m(t-1) = \{1\}^m$ starting from the same scores since $t - m$. Then, for any $t' \in T = (t, \ldots, t_r - 1)$ we have*

$$C^{MJ}(t') \subseteq C^{AG}(t') \subseteq C^{MG}(t').$$

Proof. First, we show that $C^{AG}(t) = C^{MG}(t)$ at time t. Consider an arbitrary agent i. Notice that agent i has the same score both in *AG* and in *MG*. Since $h^m(t-1) = \{1\}^m$, agent i takes the same action based on the same strategy both in *AG* and in *MG*. Thus, we have $C^{AG}(t) = C^{MG}(t)$ at time t.

Next, we show that $C^{AG}(t') \subseteq C^{MG}(t')$ at time $t' \in T$. Notice that all the agents in *MG* increase the selling strategies for $h^m(t-1) = \{1\}^m$. On the other hand, notice that the agents in *AG* that have smaller mean asset values $v(t)$ than the price $p(t)$ increase the selling strategies for $h^m(t-1) = \{1\}^m$. Since every contrarian refers to the same part (i.e., $\{1\}^m$) of the strategy, he does not change his decision during the interval T. If an agent increases the selling strategies in *AG*, it also increases the selling strategies in *MG*. Thus we have $C^{AG}(t') \subseteq C^{MG}(t')$ at time $t' \in T$.

The similar argument holds for $C^{MJ}(t') \subseteq C^{AG}(t')$. □

We call an agent a *bi-strategist* if he can take both buy and sell actions, that is, has strategies $\mathbf{R}^{i,a}$ containing both actions, for $h^m(t-1) = \{1\}^m$ or $h^m(t-1) = \{-1\}^m$. The following lemma states that there is a time lag between the price rising and the action of agent's payoff function.

Lemma 1. *In AG, suppose that a history H contains $h^m(t-1) = \{1\}^m$. Even if a bi-strategist keeps the opposite action of the payoff function for H_p steps, he takes the same action as the payoff function after the $H_p + 1$-st step.*

Proof. Suppose that a bi-strategist i has a strategy R_{i,a_1} (resp. and a strategy R_{i,a_2}) which takes the opposite action of (resp. the same action as) the payoff function. If i adopts the strategy R_{i,a_1} now, the score difference between R_{i,a_1} and R_{i,a_2} is at most $2H_p$. Since the difference decreases by 2 for a step, the scores of R_{i,a_1} and R_{i,a_2} becomes the same point at the H_p-th step. Then, after the $H_p + 1$-st step, he takes the strategy R_{i,a_2}. □

For simplicity, we assume that the size of H_p is greater than m enough.

Lemma 2. *In AG, suppose that a history H contains $h^m(t-1) = \{1\}^m$. For any time steps $t_1, t_2 \in T = (t, \dots, t_r - 1)$, where $t_1 < t_2$, we have*

$$C^{AG}(t_1) \subseteq C^{AG}(t_2).$$

Proof. Suppose that agent i belongs to $C^{AG}(t_1)$. We show that once the rising price $p(t_1)$ overtakes the mean asset value $v_i(t_1)$ of agent i, $v_i(t_1)$ will not overtake $p(t_1)$ as long as $p(t_1)$ is rising. Since

$$v_i(t+1) - v_i(t) = \frac{a(p-v)}{u+a} > 0 \text{ and } 0 < \frac{a}{u+a} < 1,$$

$p > v$ holds as long as $p(t_1)$ is rising. Thus, agent i is contrarian at time $t_1 + 1$. We have $C^{AG}(t_1) \subseteq C^{AG}(t_1 + 1)$, and can inductively show $C^{AG}(t_1) \subseteq C^{AG}(t_2)$. □

We say that the bubble is *monotone* if $h^m(t-1) = \{1\}^m$ holds for any $t \in T = (t, \dots, t_r - 1)$.

Lemma 3. *In AG, as long as more than half population are bi-strategists, the price in a monotone bubble will reach the upper bound.*

Proof. First, the mean asset values that has been overtaken by the price will not exceed the price again from the proof of Lemma 2.

Second, any bi-strategist i with $v_i(t) > p(t)$ will take a buying action in the $H_p + 1$ steps from Lemma 1. Since $v_i(t+1) - v_i(t) = a(p-1)/(u+a) < 0$, the mean asset value decreases. Thus, the rising price will eventually reach the greatest mean asset value in the set of contrarians.

Third, since all the bi-strategists increase the selling strategies, they will take selling actions in $H_p + 1$ steps. After that, $A/N < 0$ holds and the price falls down. □

From Lemma 3, the following theorem is straightforward.

Theorem 2. *In AG, as long as more than half population are bi-strategists, the monotone bubble will terminate.* □

4 Simulation

Here we present simulation results by using the basic constants in Table 1 [2].

Table 1. Basic constants

Symbol	Meaning	Value
N	Number of agents	501
S	Number of strategies	4
m	Memory size	4
H_p	Score memory	4
T	Number of steps	5000
—	Initial agent's money	10000
—	Initial agent's assets	100
r	Investment rate	0.01

Our first question with respect to ExAG is :

1. What values are suitable for the constant λ and the threshold K in ExAG ?

Our next question with respect to AG is :

2. How does the inequality of wealth distribution vary in AG ?

Then, our further questions with respect to several games are as follows.

3. How widely do the Pareto indices of games differ from practical data ?
4. How widely do the skewness / kurtosis of games differ from practical data ?
5. How widely do the volatilities differ in several games ?
6. How widely do the volatility autocorrelations differ from practical data ?

For the first issue, Figure 3 shows the patterns of price behavior for three kinds of λ values. From the definition of the direct action probability (see (11)), the smaller the λ

[2] We repeated the experiments up to 30 times and obtained averaged results.

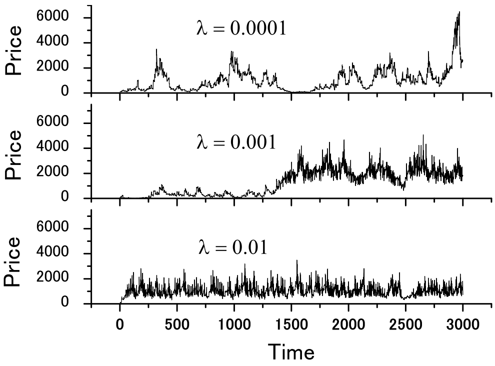


Fig. 3. Price behavior for varying λ in ExAG

becomes, the fewer the number of direct actions occur. Thus, the ratio of trend-followers is high for $\lambda = 0.0001$ and that of contrarians is high for $\lambda = 0.01$.

In addition, Figure 4 shows the skewness and the kurtosis for varying the constant λ, where the skewness (α_3) and the kurtosis (α_4) are defined as

$$\alpha_3 = \sum_{i=1}^{N} \frac{(x_i - \overline{x})^3}{N\sigma^3} \quad \text{and} \quad \alpha_4 = \sum_{i=1}^{N} \frac{(x_i - \overline{x})^4}{N\sigma^4},$$

respectively, for time series variable x_i and its average $\overline{x}$. If the skewness is negative (respectively, positive), the left (respectively, right) tail of a distribution is longer. A high kurtosis distribution has a sharper peak and longer, fatter tails, while a low kurtosis distribution has a more rounded peak and shorter, thinner tails. In other words, the more the patterns of price fluctuation occur, the smaller the kurtosis becomes. Thus, if λ is small and the reversal movements of contrarians are rare, the kurtosis becomes large. On the other hand, if we vary K^- with keeping $K^+ = 500$, the kurtosis is distributed as shown in Figure 5, where a regression curve is depicted.

From the observation above, we set $\lambda = 0.001$, $K^- = 50$ and $K^+ = 500$ in what follows.

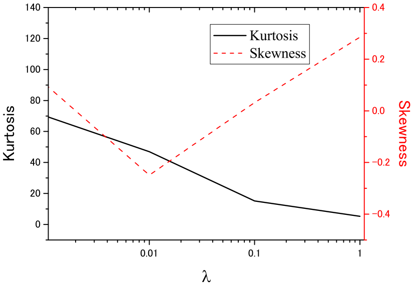


Fig. 4. Skewness / kurtosis vs λ in ExAG

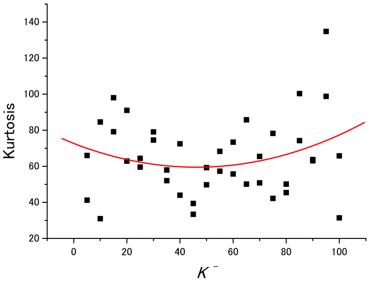
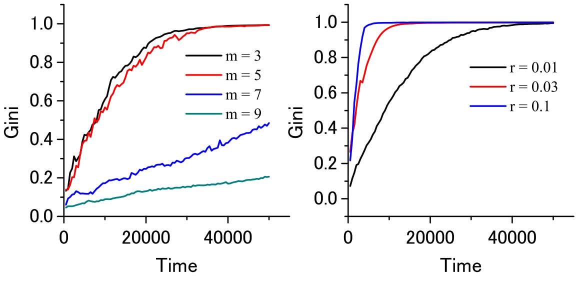


Fig. 5. Kurtosis vs K^- in ExAG



(a) Gini vs Memory Sizes (b) Gini vs Investment Rates

Fig. 6. Influence on Gini coefficient in AG

For the second issue, we present our results in Figure 6. The Gini coefficient is used as a measure of inequality of wealth distribution. Given a set of N agents' wealth $(X_1, X_2, \dots, X_N)$, the Gini coefficient G is defined as

$$G = \frac{1}{2N^2\overline{X}} \sum_{i=1}^{N} \sum_{j=1}^{N} |X_i - X_j|,$$

where $\overline{X} = \sum_{i=1}^{N} X_i/N$. If $G = 0$, the wealth is completely even. If G is close to 1, an agent has a monopoly on the wealth.

Figure 6(a) shows that the influence of memory size on the Gini coefficient. It means that the smaller the memory size is, the wider the inequality of wealth becomes. If the memory size is small, some successful agents earn much money and the others not. So their mean asset values are widely distributed in the long run. Thus, the Gini coefficient tends to be large.

Figure 6(b) shows that the influence of investment rate on the Gini coefficient. It means that the larger the investment rate is, the wider the inequality of wealth becomes. If the investment rate is large, the successful agents earn much money and the others not. So their mean asset values are widely distributed in the long run. Thus, the Gini coefficient tends to be large.

For the third issue, Figure 7 shows the price decreasing change distribution for several games and NYSE, where NYSE is the Dow-Jones industrial average 20,545 data (1928 /10/1 — 2010/7/26) in New York Stock Exchange. That is, the normalized decreasing change of price $|R| = |\Delta\text{Price}/\sigma|$ and its distribution is compared. The straight lines represent the Pareto indices. At a glance, the curves of ExAG and AG resemble that of NYSE, which means their distributions are likewise. The Pareto index of ExAG is also not far from that of NYSE.

For the fourth issue, we obtained the following results. Both ExAG and AG have better values of skewness and kurtosis than MG does as shown in Tables 2 and 3, where "stdev." and "95% int." mean standard deviation and 95% confidence interval, respectively.

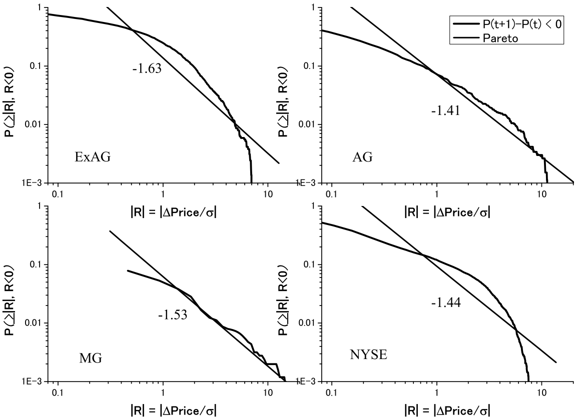


Fig. 7. Pareto indices for several games and NYSE

Table 2. Skewness

method	ExAG	AG	MG	NYSE
average	0.098	0.39	-0.32	3.725
stdev.	1.82	1.03	1.86	—
95% int.	[-0.58,0.77]	[0.007,0.77]	[-1.02,0.37]	—

Table 3. Kurtosis

method	ExAG	AG	MG	NYSE
average	42.3	72.9	148	18.92
stdev.	61.6	96.3	231	—
95% int.	[21.3,67.3]	[36.9,109]	[62.2,235]	—

For the fifth issue, we present our results in Figure 8. The volatility is defined as the standard deviation of the number of excess demand. The figure shows that the volatility of AG is lower than other games for every memory size. This means the memory size does not have a great impact on the price formation in AG.

For the sixth issue, the autocorrelation function $C(\tau)$ is defined as

$$C(\tau) = \frac{\langle A(t)A(t+\tau)\rangle}{\langle A(t)^2\rangle},$$

where τ is a time lag. The value of $C(\tau)$ becomes 1 (respectively, -1) if there is a positive (respectively, negative) correlation between $A(t)$ and $A(t+\tau)$. As shown in Figure 9, only MG has the alternating, strong positive/negative correlation for every time lag. Other games, AG and ExAG, have weak correlations which reduce as the time lag grows. The practical data, NYSE, has a negative correlation only when the time lag is $\tau = 1$. Since the excess demand in NYSE is unknown, we assume the number of

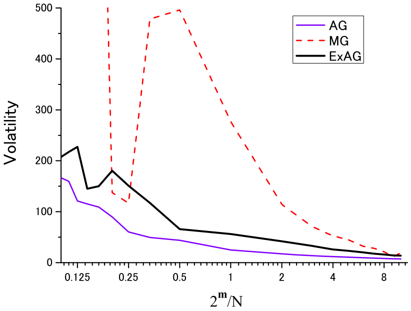


Fig. 8. Volatility ($N = 51 \sim 5119$, $S = 4$, $m = 9$)

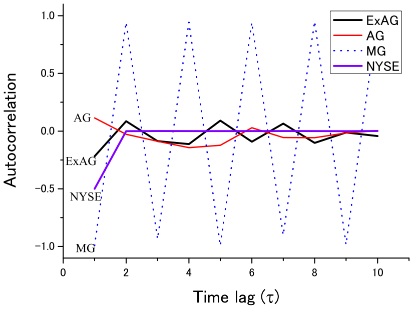


Fig. 9. Autocorrelation of volatility

agents is equal to $N = 501$ and estimate $A(t)$ from the equation (7). Notice that ExAG has the same (negative) correlation as NYSE when $\tau = 1$, while AG has the positive correlation.

5 Conclusions

In this paper, we proposed an asset value game and an extended asset value game. The AG is a simple variant of MG such that the only difference is their payoff functions. Though the AG captures a good feature of an agent's behavior, there is a time lag between the rapid change of a price and the adjustment of an agent's behavior. So we consider the ExAG, an improvement of AG, by using parameters which contain some probabilistic behavior. The ExAG has two parameters by which the balance of trend-followers and contrarians can be controlled. We examined several values for the parameters and then fixed to specified values. We obtained several experimental results which reveals some characteristics of ExAG. The advantages of ExAG are twofold. First, we can restrict a drastic movement of price in AG by tuning the parameters. Second, we can reduce the time lag generated by recovering score losses in AG.

Our future work includes investigating the influence of market intervention, an in-depth analysis of the AG, and other applications of the games.

Acknowledgements. This work was partially supported by Grant-in-Aid for Scientific Research (C) 22510147 of the Ministry of Education, Science, Sports and Culture of Japan.

References

1. Andersen, J.V., Sornette, D.: The \$-game. The European Physical Journal B 31(1), 141–145 (2003)
2. Araujo, R.M., Lamb, L.C.: Towards understanding the role of learning models in the dynamics of the minority game. In: Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004), pp. 727–731 (2004)
3. Araujo, R.M., Lamb, L.C.: On the use of memory and resources in minority games. ACM Transactions on Autonomous and Adaptive Systems 4(2), 2 (2009)
4. Challet, D.: Inter-pattern speculation: beyond minority, majority and \$-games. Journal of Economic Dynamics and Control (2007)
5. Challet, D., Marsili, M., Zhang, Y.-C.: Stylized facts of financial markets and market crashes in minority games. Physica A 294, 514–524 (2001)
6. Challet, D., Marsili, M., Zhang, Y.-C.: Minority games — interacting agents in financial markets. Oxford University Press (2005)
7. Challet, D., Zhang, Y.-C.: Emergence of cooperation and organization in an evolutionary game. Physica A 246, 407–418 (1997)
8. Coolen, A.C.C.: The mathematical theory of minority games. Oxford University Press (2005)
9. Ferreira, F.F., Marsili, M.: Real payoffs and virtual trading in agent based market models. Physica A 345, 657–675 (2005)
10. Ferreira, F.F., de Oliveira, V.M., Crepaldi, A.F., Campos, P.R.A.: Agent-based model with heterogeneous fundamental prices. Physica A 357, 534–542 (2005)
11. Giardina, I., Bouchaud, J.-P., Meźard, M.: Microscopic models for long ranged volatility correlations. Physica A 299, 28–39 (2001)
12. Kiniwa, J., Koide, T., Sandoh, H.: Analysis of price behavior in lazy \$-game. Physica A 388(18), 3879–3891 (2009)
13. Liu, X., Liang, X., Tang, B.: Minority game and anomalies in financial markets. Physica A 333, 343–352 (2004)
14. Marsili, M.: Market mechanism and expectations in minority and majority games. Physica A 299, 93–103 (2001)
15. De Martino, A., Giardina, I., Mosetti, G.: Statistical mechanics of the mixed majority-minority game with random external information. Journal of Physics A: Mathematical and General 36, 8935–8954 (2003)
16. Takayasu, H., Miura, T., Hirabayashi, T., Hamada, K.: Statistical properties of deterministic threshold elements — the case of market price. Physica A 184(1-2), 127–134 (1992)
17. Tedeschi, A., De Martino, A., Giardina, I.: Coordination, intermittency and trends in generalized minority games. Physica A 358, 529–544 (2005)
18. Yamada, K., Takayasu, H., Takayasu, M.: Characterization of foreign exchange market using the threshold-dealer-model. Physica A 382(1), 340–346 (2007)

A Strategy for Efficient Persuasion Dialogues

Katie Atkinson[1], Priscilla Bench-Capon[2], and Trevor Bench-Capon[1]

[1] Department of Computer Science, University of Liverpool, U.K.
[2] Open University, Milton Keynes, U.K.
{katie,tbc}@liverpool.ac.uk, pbench-capon@gmail.com

Abstract. Several dialogue types, including inquiry, persuasion and deliberation, transfer information between agents so that their beliefs and opinions may be revised. The speech acts in different dialogue types have different pragmatic implications. For a representative sub-type of persuasion dialogues we consider how they can be conducted efficiently, in terms of minimising the expected transfer of information, and develop a strategy for efficient persuasion by exploiting the pragmatic implications. We demonstrate that our strategy is optimal for this sub-type.

1 Introduction

Several authors have developed dialogue protocols for use in agent systems based on the typology of Walton and Krabbe [9], including [2] for inquiry dialogues; [6] for persuasion; and [5] for deliberation. It is argued in [9] that, unlesss agents recognise (and agree on) the type of dialogue they are engaged in, misunderstandings arise, fallacies become possible and the conversation may break down. Even so, the distinctions have rarely been made precise, and confusion is increased because inquiry, persuasion and deliberation all make use of a similar set of speech acts. In this paper we will give a precise characterisation of the distinctive features of a sub-type of persuasion dialogues and distinguish them from deliberation and inquiry dialogues. Our analysis will draw attention to the pragmatic meaning of utterances which vary across dialogue types and give them their distinctive character. We offer a clear specification of these persuasion dialogues, and an optimal strategy for conducting a dialogue of this representative sub-type. Our contribution is thus the analysis yielding the distinguishing features of persuasion dialogues, and an optimal strategy for a common class of persuasion dialogues.

One common situation that gives rise to both persuasion and deliberation is when an agent must choose between several options, each of which has several features that can serve as reasons for and against the choice, and to which agents will ascribe different degrees of importance depending on their personal preferences. When buying a car one person will be most interested in safety, another in speed, another in comfort. For cameras, some will require the best quality, whereas others will value more highly compactness or a low price. In such situations an agent will often need to deal with a series of salespeople, each trying to overturn the agent's currently preferred option, or to consider recommendations from other agents. A very common example in AI conference papers is choosing a restaurant for lunch or an evening dinner. Typically the individual agents will have only incomplete and often ill-founded or outdated beliefs

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 332–347, 2013.

about the local eateries, and so they will pool their knowledge to get a more complete picture of the situation before deciding. Thus one agent may solicit recommendations, and another agent may attempt to persuade that agent of its own favourite venue. We will use this illustration throughout this paper.

In [9], dialogue types are characterised by an initial situation, a collective goal, and individual goals, all stated informally. For inquiry dialogues the initial situation is that both agents are not certain of some proposition p. Both individual goals and the collective goal are the same: to determine whether or not p. In persuasion one agent will argue that p, or that some action ϕ should be done hoping that the other agent will come to agreement. The collective goal is to resolve whether p is true or ϕ should be done. With regard to individual goals, persuasion is asymmetric: the persuader wishes to convince the persuadee, whereas the persuadee wishes to explore the possibility that its current opinion should be revised in the light of information known to the persuader: the persuadee is interested in what is true, whether it be p or $\neg p$. A different case of persuasion is what Walton terms a *dispute* [8]. In this case the persuadee also wishes to convince the other agent that its own original position is correct, so that its individual goal is now that the other should believe $\neg p$ or that ϕ should not be done: we will not consider disputes further in this paper. Deliberation is generally held to concern actions: initially both agents are unsure whether or not to ϕ, and individually *and collectively* they wish to come to agreement as to whether or not to ϕ. In the next section we will explore the distinctions further, with a view to precisely characterising persuasion dialogues in particular.

2 Distinguishing the Dialogue Types

Although some have suggested that persuasion cannot be applied to actions, this is contrary to common usage and we believe that the correct distinction is related to *directions of fit*, a distinction made by e.g. Searle [7]. Searle distinguishes theoretical reasoning, reasoning about what is the case, from practical reasoning, reasoning about what it is desired to be the case, and what should be done to realise those desires. In the first case it is necessary to fit one's beliefs to the world, whereas in the second the idea is to make the world fit one's desires, in so far as one has the capacity to do so. In these terms, inquiry represents an attempt to better fit the beliefs of the agents to the world, and deliberation how best to make the world fit the collective desires of the agents. Persuasion can be about either. Note, however, that when we have two (or more) participating agents, we have two (or more), probably different, sets of desires to consider. In deliberation no set of desires should be given pre-eminence, but rather the group as a whole needs to come to an agreement on what desires they will adopt collectively (e.g. [5]). In contrast, as discussed in [1], in persuasion it is the desires of the *persuadee* that matter: a persuadee is fully entitled to use its own preferences to assess any proposition or proposal, without any need to consider what the persuader desires. The construction of a set of collective desires introduces an additional order of complication, and puts deliberation beyond the scope of this paper. Therefore in what follows we will focus exclusively on persuasion.

Essentially an agent starts with incomplete information. In a dialogue agents can pool information that can extend their knowledge and perhaps revise their beliefs. At any time an agent has a belief about the utility of various options, but this is tentative because of the incompleteness and potential incorrectness of its beliefs. The purpose of persuasion is thus to acquire information and verify the current beliefs to confirm the current preference or establish a new one. In a deliberation not only is information about facts exchanged, but also information about the importance of particular attributes for particular agents in determining the utility of an option. Thus an inquiry concerns a particular item of information. A persuasion concerns information relating to the set of attributes relevant to determining a particular preference. A deliberation concerns this information, and information as to what attributes might be used to determine this preference. Persuasion and deliberation can thus be seen as composed of a series of inquiry dialogues, with the context determining the subjects of inquiry.

2.1 Definitions

An example, which illustrates our notation, is given in Section 3. The reader might find it helpful to refer to this in conjunction with the following definitions for concrete examples of their use.

The knowledge bases of agents can be partitioned into *factual elements*, on which agents should agree, used when the direction of fit is from world to beliefs, and *preference elements*, which represent their own individual desires, tastes and aspirations, and are used when the direction of fit is from desires to the world. Thus, the preference elements represent the way the agent evaluates possible options to determine which wishes to bring about, and how it evaluates objects and situations for value judgements such as *best car* and *acceptable restaurant.*

Definition 1. *Let AG denote a set of agents, each of which, $Ag \in AG$, has a knowledge base KB^{Ag}. KB^{Ag} is partitioned into factual elements denoted by KB_F^{Ag} and preference information denoted by KB_P^{Ag}. KB_F^{Ag} comprises facts, strict rules and defeasible rules. KB_P^{Ag} comprises rules to determine the utility for Ag of certain items based on their attributes, and the weights used by these rules. These preference elements are defined below.*

Agents expand their KB_F by taking information from one another, but KB_P remains fixed throughout the dialogue. Whereas, because it is intended to fit the world, KB_F is objective, KB_P represents the personal preferences of an individual agent, and so is entirely local to the agent concerned. We will use f for factual propositions, and p_{Ag} (to be read as "p is the case for Ag") for propositions based on preferences. We will not represent actions separately, so that p_{Ag} may represent either propositions such as *Roux Brothers is an acceptable restaurant for Ag* or propositions such as *it is desirable for Ag_A that Ag_B ϕ.*

Definition 2. *Let $PROP$ be a set of atomic propositions and let $f \in PROP$ be a* factual proposition *and $p_{Ag} \in PROP$ be a proposition based on preferences. Let $KB_F^{Ag} \rightsquigarrow f$ denote that a factual proposition can be defeasibly shown from KB_F^{Ag}, using an appropriate reasoning engine (e.g. [2] uses Defeasible Logic [3]), without using* *preference information.* KB_P^{Ag} *records the preferences of Ag using clauses with preference based propositions as heads and bodies comprising factual propositions. Now* $KB_F^{Ag} \cup KB_P^{Ag} \rightsquigarrow p_{Ag}$ *denotes that the preferences of Ag are required for* p_{Ag} *to be shown.*

This reflects that a value judgement such as *Rolls Royce make the best cars*, cannot be considered true *simpliciter*, but is true (or false) relative to an agent, and determined using that agent's individual preferences. Next we may need to distinguish between the knowledge base of an agent at the start of the dialogue, and the knowledge base of that agent at the end of the dialogue. In our simple case, only KB_F^{Ag} changes during a dialogue. While an agent may be persuaded to change its preferences, this is a higher order persuasion in this paper and not a straightforward matter of simply adding to the existing KB_P, as is the case with facts.

Definition 3. *Let the knowledge base of an agent Ag at the start of a dialogue be denoted by* KB^{Ag_0}*, and its knowledge base after n steps of dialogue be denoted by* KB^{Ag_n}.

We now introduce some further definitions needed for the protocol and strategy. Firstly, agents need to be able to discuss the options under consideration within the dialogue and the attributes of these options. For example, in a dialogue about where to go on holiday, the options might cover different countries such as France and Spain with relevant attributes including things such as the food and the weather. In the restaurant example the various local restaurants will be the options and various features such as distance, quality, type of cuisine, ambiance and the like are attributes of restaurants that agents may use as reasons to choose between them. Furthermore, agents can give individual weightings to these attributes to reflect their different tastes and aspirations.

Definition 4. *Let O be a set of options that agents can propose during the course of a dialogue. O has an associated set of attributes* A_O*. An agent associates each* $a_j \in A_O$ *with a weight* $w_j \geq 0$ *to form a set of weights W. Each set W is individual to an agent.*

Next we need to be able to determine the truth value of attributes of options, e.g. stating that the option Spain does indeed have the attribute of good weather, or that the Mogul Palace serves Indian food. We also need to be able to determine the weight that an agent assigns to an attribute of an option. The following two functions enable the above.

Definition 5. *Let* $\tau : O \times A_O \rightarrow \{0, 1\}$ *be a truth function that returns the truth value given by* $\tau(O_i, a_j) = \tau_{ij}$ *for option* $O_i \in O$ *and attribute* $a_j \in A_O$.

Definition 6. *Let* $w : AG \times A_O \rightarrow \mathbb{N} \cup \{0\}$ *be a weight function that returns the weight* $w(Ag, a_j) = w_{Ag}(a_j)$ *of an attribute* a_j *for agent* $Ag \in A_O$*. Where the agent Ag is clear from the context, we use* w_j.

Next we introduce notation to enable us to refer to sets of attributes of options as determined by their truth status (as will be required in section 4). Thus, attributes of options will fall within one the following disjoint sets: *verified true*, *verified false*, *unverified* *true* and *unverified false*, where verified options have been the subject of an inquiry dialogue (and so have been agreed by the agents) and unverified options have not (and so could change in the light of information that could be elicited by an inquiry dialogue).

Definition 7. *τ_{ij} = 1 if option O_i has attribute a_j. If this attribute for this option has been the subject of an inquiry dialogue, τ_{ij} has been verified. Attributes of options for which τ_{ij} = 1 has been verified form a set VT_{ag}; those for which τ_{ij} = 0 has been verified form a set VF_{ag}, for both agents. For a particular agent, if for an unverified attribute $KB_F^{Ag} \rightsquigarrow \tau_{ij} = 1$, the attribute is unverified true and so an element of UT_{ag}: otherwise the attribute is unverified false and so is an element of UF_{ag}. $VT_{ag} \cup UT_{ag} \cup VF_{ag} \cup UF_{ag}$ = A_O.*

Finally, we are able to define the *utility* of an option for an agent, based on attributes of the option that are true.

Definition 8. *Let $A \subseteq A_O$ be a set of attributes true of O_i. Then the utility of O_i for an Agent Ag with respect to these attributes, $U_i(A)$, is $\sum_{a_j \in A} w_{Ag}(a_j)$. Now the current utility is $U_i(VT_i \cup UT_i)$. This we will sometimes abbreviate to U_i when there is no ambiguity. The maximum utility is $U_i(VT_i \cup UT_i \cup UF_i)$ and the minimum utility is $U_i(VT_i)$.*

2.2 Inquiry and Persuasion Dialogues

We can now characterise the distinction between inquiry and persuasion. Suppose we have two agents, Wilma and Bert, so that $AG = \{W, B\}$: following the conventions of chess, W will initiate the dialogues. We may say that for an inquiry dialogue between Wilma and Bert concerning a proposition f (inquiry dialogues concern *only* factual propositions), the following two conditions should hold:

I1 Initially either Wilma does not believe that f: $KB_F^{W_0} \not\rightsquigarrow f$ or Bert does not believe that f : $KB_F^{B_0} \not\rightsquigarrow f$ or both.

I2 At the end of the dialogue both believe that f if and only if f is a consequence of the joint knowledge bases: $(KB_F^{W_1} \rightsquigarrow f) \wedge (KB_F^{B_1} \rightsquigarrow f) \leftrightarrow KB_F^{W_0} \cup KB_F^{B_0} \rightsquigarrow f$.

An inquiry dialogue will always result in agreement, since, if explicit agreement does not terminate the dialogue it will continue until Bert and Wilma have exchanged all their knowledge. Thus, if the dialogue has taken *n* steps, $KB_F^{W_n} = KB_F^{B_n} = KB_F^{W_0} \cup KB_F^{B_0}$. Some have argued that neither should believe that f at the outset, but we wish to allow Wilma to start an inquiry dialogue to confirm her beliefs, since, given the defeasible reasoning mechanism we are using, it is always possible that Bert may supply information resulting in Wilma revising her beliefs.

The second condition is plausible (since we are assuming that factual propositions are objectively assessed by the agents) and is the condition used to show soundness and completeness of the inquiry dialogues in Black and Hunter [2]. Since that paper shows soundness and completeness for their inquiry dialogues, we shall suppose that where

Wilma and Bert wish to establish the truth of some factual proposition, they will use a dialogue as described there.

In contrast, persuasion can concern matters with both directions of fit, although probably evaluative propositions are more usually the topic than factual ones. This is because if the persuader is correct (and honest), for factual topics an inquiry dialogue will serve to achieve the goals of a persuasion dialogue. It may be, however, that the persuader has some interest in establishing the proposition, and this interest will persist even if it becomes aware of information suggesting that the proposition is in fact false. If one of the agents represents a salesperson, for example, it may act in this way. Since persuasion may have this adversarial element we distinguish *open persuasion*, where the persuader is expected to tell the truth, the whole truth and nothing but the truth, and *partial* (i.e not *impartial*) *persuasion* where the persuader must tell only the truth and nothing but the truth, but need not tell the whole truth if that would damage its cause. Open persuasion about a fact is simply an inquiry dialogue where the persuader initially believes the proposition under consideration, so that these dialogues can also be characterised by I1 and I2 above. We will henceforward confine ourselves to persuasion dialogues that concern matters of preference, so that there will be subjective elements dependent on tastes and preferences as well as facts.

Open Persuasion
OP1: Wilma believes that Bert does not believe that p_B: $KB^{B_0} \not\rightsquigarrow p_B$
OP2: Both wish that at the end of the dialogue Bert believes that p_B if and only if p_B is a consequence of their joint knowledge bases and Bert's preferences: $KB^{B_t} \rightsquigarrow p_B \leftrightarrow KB_F^{W_0} \cup KB^{B_0} \rightsquigarrow p_B$, where the dialogue terminates in *t* steps.

Note that at the end of an open dispute, Bert and Wilma may differ as to whether p or $\neg p$, since their beliefs reflect their own individual preferences applied to the shared information. Again the dialogue will always terminate because either the agent is persuaded, or all factual knowledge has been exchanged.

For partial persuasion we include the desire to, as it were, win the dialogue, irrespective of the truth of the matter. Although partial persuasion can be conducted regarding a matter of fact, we will consider here only persuasion relating to matters involving elements of preference. The initial goal remains OP1, and Bert's final goal remains OP2, but Wilma has a different goal, PP3. PP4 represents the condition under which Wilma can legitimately satisfy her goal. Again the dialogue terminates in *t* steps.

Partial Persuasion using Preferences
PP3: Wilma's goal is that Bert should believe that p: $KB^{B_t} \rightsquigarrow p_B$
PP4: Wilma can succeed if she has information to enable Bert to believe p: $KB^{B_t} \rightsquigarrow p_B \leftrightarrow (\exists S) \subseteq KB_F^{W_0}$ such that $S \cup KB^{B_0} \rightsquigarrow p_B$.

Note that open persuasion begins with a conflict, but has the same goals for both participants, whereas in partial persuasion they have different views as to what constitutes a successful termination. Both agents can realise their individual goals if p_B does follow from their collective knowledge and Bert's preferences. If not, strategic considerations may affect the outcome: it may be that there is a subset of Wilma's KB which could be used to persuade Bert, but that she reveals too much, so that Bert is unpersuaded. Here

Wilma could have succeeded according to PP4, but in fact Bert is rightly unpersuaded, satisfying OP2.

Having made these distinctions we will now focus on a particular sub-type of persuasion dialogue, open persuasion involving preferences. This is the simplest kind of dialogue after inquiry dialogues, in that only one side is doing the persuasion, and the participants have common goals. Our particular scenario is the common situation where an agent is seeking a recommendation, or comparing options, concerning things such as restaurants, cars, digital cameras, insurance, or any other product where competing options have some out of a large number of features that vary in importance for different agents. The persuading agent will want what is best for the other agent, and will have no particular interest in having its recommendation accepted. The particular topic we will consider is *what is an acceptable restaurant for Bert*?

3 Example

For our example we will spell out choice of restaurant in detail. Let us suppose that Wilma and Bert are standing outside Burger World. Bert can see that Burger World is close by and appears cheap, but as a stranger to the town knows nothing about any other restaurant. Wilma, on the other hand, as a native to the city, has complete information. So, initially, Bert finds Burger World acceptable but Wilma will attempt to persuade him of the merits of the Thai Palace. Moreover suppose Bert wants a good quality restaurant that is cheap, close by and licenced. Bert weights these attributes 6,2,1 and 2 respectively. All other features of restaurants, such as whether they have music, and the type of cuisine, are matters on which Bert is indifferent, and so have weights of 0, and need not be considered. We can summarise the situation:

Set of Agents AG = {Wilma, Bert}. Wilma is the persuader and Bert is the persuadee.

Set of Options O = {BurgerWorld, ThaiPalace} = $\{O_1$, $O_2\}$

Set of Attributes A_O = {goodQuality, cheap, close, licenced} = $\{a_1, a_2, a_3, a_4\}$

Sets of weights. $W_{Wilma(}a_i)$ = {6,4,0,2}; $W_{Bert}(a_i)$ = {6,2,1,2};
for i = 1,2,3,4.

Truth values τ_{ij}. Burger world $(j = 1)$ {0,1,1,0} (cheap and close). Thai Palace $(j = 2)$ {1,1,0,1}: (good quality, cheap and licenced).

Each agent, for each option, partitions A_O into four subsets, depending on its own knowledge base. Attributes that are not known and have not yet been the subject of an inquiry are *unverified*, while those that have been the subject of an inquiry are *verified*, and the agents are in agreement as to them. If an attribute cannot be shown true, it is considered to be false since the defeasible reasoner uses negation as failure. So at the start of the dialogue,

Bert has $UT_1 = \{cheap, close\}, UF_1 = \{good, licenced\}, VF_1 = VT_1 = \emptyset$ for Burger World and $UF_2 = \{good, cheap, close, licenced\}, UT_2 = VF_2 = VT_2 = \emptyset$ for the Thai Palace. As the dialogue progresses, inquiries regarding attributes are made, and these attributes will move from unverified to verified. Bert's utility calculations for Burger World at the start of the dialogue are shown in Figure 1.

W1 I would go to the Thai palace. *Wilma starts the persuasion dialogue by making a recommendation.*

B1 Burger World is right here and it looks cheap. *Bert indicates two criteria that he values and that he believes are satisfied by Burger World*

W2 The Thai Palace is also cheap, but it is a walk away. *Wilma supplies information about the criterion satisfied by the Thai Palace. Bert increases the current and minimum utilities of the Thai Palace to 3.*

B2 Is Burger World good? *Bert seeks information about another valued criterion.*

W3 No. But the Thai Palace is. *Wilma indicates a point in favour of the Thai Palace. Bert must now adjust his utilities: while the current utility of Burger World remains 3, the maximum falls to 5. But the minimum utility of the Thai Palace is now 6, and so cannot be bettered by Burger World.*

B3 OK. *Bert now has sufficient information about both restaurants for the criteria he values: he does not ask about licensing because that can no longer change the order for him.*

At the end of the dialogue: Bert has $UT_1 = \emptyset, UF_1 = \{licenced\}$, $VF_1 = \{goodQuality\}, VT_1 = \{cheap, close\}$ for Burger World and $UT_2 = \emptyset, UF_2 = \{licenced\}, VF_2 = \{close\}, VT_2 = \{cheap, goodQuality\}$ for the Thai Palace. The utility calculations for Bert and Burger World at the end of the dialogue are shown in Figure 1.

Calculation of Utilities for Bert for Burger World at the start of the dialogue:

Current Utility = $U_1(UT_1 \cup VT_1))$ = $U_1(\{cheap, close\})$ = $w_B(a_2) + w_B(a_3)$ = $2 + 1$ = 3.

Minimum Utility = $U_1(VT_1))$ = 0

Maximum Utility = $U_1(UT_1 \cup VT_1 \cup UF_1))$ = $U_1(\{goodQuality, cheap, close, licenced\})$ = $6 + 2 + 1 + 2$ = 11.

Calculation of Utilities for Bert for Burger World at the end of the dialogue:

Current Utility = $U_1(UT_1 \cup VT_1))$ = $U_1(\{cheap, close\})$ = $w_B(a_2) + w_B(a_3)$ = $2 + 1$ = 3.

Minimum Utility = $U_1(VT_1))$ = 3

Maximum Utility = $U_1(UT_1 \cup VT_1 \cup UF_1))$ = $U_1(\{cheap, close, licenced\})$ = $2 + 1 + 2$ = 5.

Fig. 1. Utility Calculations for Bert

This is a fairly efficient dialogue: restaurants have many attributes and so Wilma could have told Bert many things he did not know, whereas the dialogue is able to conclude after Bert has received just six items of information. How is this possible? It is because Wilma is able to infer things about Bert's criteria and current knowledge beyond what Bert explicitly states and asks, and so can recognise what will be relevant to Bert's opinion. This is Grice's notion of *conversational implicature*. In [4], Grice advanced four maxims (Quality, Quantity, Relevance and Manner) intended to capture the pragmatics of utterances in cooperative dialogues. The maxims express the cooperation principle:

> Make your conversational contribution such as is required, at the stage at which it occurs, by the accepted purpose or direction of the talk exchange in which you are engaged,

which arguably must be observed if misunderstandings and conversational breakdown are to be avoided. Relevance (which may be dependent on the dialogue type and its state) is our main concern here.

Thus Bert begins in B1 by stating that Burger World satisfies two criteria. Burger World is not chosen at random but is Bert's currently preferred option, the one which the Thai Palace must overcome. The criteria are those which Bert believes Burger World does, and the Thai Palace does not, satisfy. Thus in Wilma's reply she can improve her case by stating that one of the criteria is satisfied by the Thai Palace. Once a criterion is put into play, it is considered for both options, thus verifying any existing beliefs Bert may have about the options. In B2 Bert asks a question. This again is not chosen at random but concerns a criterion which, if satisfied, would put Burger World beyond reach. Wilma can, however, truthfully say that it does not satisfy this criterion, but that the Thai Palace does. Bert now draws the dialogue to a close since he has sufficient knowledge of the criteria relating to the two options relevant to him, according to his preferences. He does not ask about licencing because that cannot restore Burger World's lead. In this dialogue features of the Thai Palace that make it attractive to Wilma are not even mentioned, while she supplies information about a feature to which she is indifferent: Bert is the sole arbiter of what makes the restaurant good (in complete contrast to deliberation). Bert could not ask about another criterion without misleading Wilma by *conversationally implying* that he valued a fourth criterion enough to overturn his current view, which would prolong the dialogue to no useful purpose. Note also the asymmetry: Wilma's criteria may determine her recommendation, but play no part in the dialogue unless they are shown to be valued by Bert.

The dialogue is not, however, as efficient as it might be. Given his weights, if Bert discovered that either restaurant was, and the other was not, of good quality, he could stop immediately, since this criterion carries more weight than the other three combined. Thus Bert's best initial question would have been B2, with Wilma's reply in W3 enough to resolve the discussion. We could extend our notion of conversational implicature to suggest that mentioning a criterion means not only that it matters to the persuadee but additionally that there are no more important criteria not yet mentioned. This is what we will do in the algorithm developed in the next section. As we will see in section 5, this algorithm, based on conversational implicature, is optimal in terms of minimising the expected exchange of information.

Note also that Bert's strategy is good only with regard to open persuasion. Had this been a partial persuasion situation, Bert's question in B2 could only have received the response that the Thai Palace was good. Since Wilma is not obliged to tell the whole truth, even if she had believed that Burger World was good, she would have remained silent on that point, and indeed on all positive features of Burger World. Wilma answers the question only if the answer does not improve Bert's assessment of that restaurant. In a partial persuasion, therefore, Bert will ask only whether the Thai Palace is good.

If Bert does not know some relevant facts about Burger World, he should attribute some kind of expected utility to them, in which case discovery that the Thai Palace was of good quality might not immediately displace Burger World, and the question as to price would still be important.

Beneath the surface of the dialogue, the pragmatic meaning of the utterances in the context of persuasion is that Bert is asking a sequence of questions about whether attributes that he values enough to be able to change his current preference are satisfied by the proposed option and (in open persuasion) by his currently preferred option. Wilma's role is to answer these questions. We suggest also that Bert should ask about criteria in order of their importance to him. We now develop an algorithm based on this principle and show in section 5 that this represents an optimal strategy. Thus following Grice's maxims, and taking the pragmatics of the dialogue type seriously, has computational benefits.

4 Protocol and Strategy

The persuasion dialogue, (W1-B3 in the example), once it has been initiated by the persuader, is a series of statements and requests for information on the part of the persuadee. Either these will be a statement of the form $\tau_{1j} = 1$, or a question of the form whether $\tau_{1j} = 1$. In either case, in the terms of [2], the persuadee wishes to open inquiry dialogues with τ_{1j} and τ_{2j} as topics (in open persuasion) or with τ_{2j} as the only topic (in partial persuasion).

This can be accomplished using a series of inquiry dialogues. Since that of [2] has been shown to be sound and compete, it can be used to give Bert the best information available, the knowledge derivable from the union of the KBs. We will therefore assume the use of the protocol of [2] to exchange information. When should the persuasion dialogue terminate? If O_1 is Bert's currently best option, and Wilma is trying to persuade him of O_2, Bert should continue to seek information until the minimum utility of O_1 is greater than the maximum utility of O_2, in which case Wilma's proposal can be rejected, or until minimum utility of O_2 exceeds the maximum utility of O_1, in which case her proposal can be accepted. This state will be approached by initiating inquiry dialogues to verify as yet unverified features of the options. When $UT_i = UF_i = \emptyset$, the dialogue must terminate, since the option with the highest current utility cannot be displaced. Often, however, it will terminate with only a subset of the attributes verified: in the example above it may terminate if good quality is true of one but not the other; if true of both or neither it could terminate for Wilma if cheap is true of one but not the other, and she will consider whether they are licenced only if the options are the same in terms of quality and price.

Definition 9. *Termination condition. Let $\mathcal{T}_1(X)$ be the condition that the algorithm has terminated with O_1 preferred when the status of all the attributes associated with the weights in $X \subseteq W$ is known to the persuadee.*

$\mathcal{T}_1(X)$ holds if and only if
$\sum_{w_j \in X} \tau_{1j} w_j - \sum_{w_j \in X} \tau_{2j} w_j > \sum_{w_j \in W \setminus X} w_j$. *Similarly for $\mathcal{T}_2(X)$.*

For efficiency, the agent should choose the topics so as to minimise the expected number of steps. From our observations about what is conversationally implied in persuasion dialogues, we conjecture that an optimal strategy, $\mathcal{S}$ for an agent would be to inquire about attributes in descending order of their weights, and so employing a kind of greedy algorithm. We now present the protocol and embedded strategy $\mathcal{S}$ for our persuasion dialogue. Agents can *open* and *close* the relevant dialogues, *propose* options and *inquire* about the status of attributes, as in [2]. A persuasion dialogue follows the following protocol:

> [P0]: Wilma opens by proposing an option, O_2. If Bert's currently preferred option is $O_1 \neq O_2$ a persuasion dialogue will commence; otherwise he agrees immediately. Initially $A = X = \emptyset$; W = the set of weights of all attributes about which Bert may inquire.
> [P1]: Bert opens inquiry dialogues with topics τ_{1j}, τ_{2j} for some attribute a_j with which Bert associates a positive weight w_j; A becomes $A \cup \{a_j\}$; X becomes $X \cup \{w_j\}$; increment the utilities $U_i : i \in \{1, 2\}$ by $\tau_{ij} w_j$. This may change which option is currently preferred.
> [P2]: If $\mathcal{T}_1(X)$ holds terminate with O_1 preferred, else if $\mathcal{T}_2(X)$ hold terminate with O_2 preferred else if $X = W$, return the currently preferred option; else go to [P1].

The inquiry about a single attribute in [P1] and [P2] is termed a *step*. Moves are subscripted with W or B depending on whether the move is made by the agent acting as the persuader (W) or persuadee (B). There are two options O_1 and O_2. Thus the dialogue begins with W proposing O_2. Now B either agrees, or states its preferred option and inquires about some attribute. Since only B's weights matter, w_j will refer to the weight given to a_j by B.

```
open dialogue_W
propose_W(O_2)
if agree_B(O_2))
  $\mathcal{T}_2(\emptyset)$, end dialogue_B
else
  $X = \emptyset$; $A = \emptyset$
  propose_B(O_1), such that $O_2 \neq O_1$
  open persuasion dialogue_B
  for all $a_j \in A_O$ do
    sort $A_O$ into ordered descending list such that $w_j \geq w_{j+1}$
  end for
  $j$=1
  while $\neg\mathcal{T}_1(X)$ and $\neg\mathcal{T}_2(X)$ do
    inquire_B($\tau_{1j}, \tau_{2j}$)
    $A$ becomes $A \cup \{a_j\}$; $X$ becomes $X \cup \{w_j\}$;
    increment the utilities $U_i : i \in \{1, 2\}$ by $\tau_{ij} w_j$.
    if $\tau_{1j}$ = 1 then $a_j \in VT_1$ else $a_j \in VF_1$
    end if
    if $\tau_{2j}$ = 1 then $a_j \in VT_2$ else $a_j \in VF_2$
    end if
```

```
        j++
    end while
    if $\mathcal{T}_1(X)$ then disagree_B(O_2) else agree_B(O_2)
    end if
end if
end dialogue_W
```

The algorithm given above is for open persuasion; for partial persuasion the inquiries will concern τ_{2j} only. As noted above, each pass through the while loop is referred to as a *step* in the remainder of the paper. Each step establishes or verifies the truth values for one attribute.

5 Results

In this section we describe a series of results that show our strategy $\mathcal{S}$ to be an optimal strategy for a number of representative situations. The first of these is where the persuadee has no initial opinions as to the facts: the attributes of the options will be discovered from the dialogue, and are all considered equally likely at the outset. Theorem 1 shows that $\mathcal{S}$ is optimal for this scenario[1].

Theorem 1: Suppose we have two options O_1 and O_2 of equal prior utility, m attributes, which are equally likely to be true or false for each option, and m positive weights assigned to the attributes. Then an optimal strategy, in the sense that the algorithm terminates in the expected fewest number of steps, is to inquire about the attributes in descending order of weight (strategy $\mathcal{S}$). First we prove two preliminary lemmas.

Lemma 1: Given a set A of n attributes, the probability that the algorithm will terminate in not more than n steps is independent of the order in which we inquire about these attributes. We term this probability $P(X)$ where $X = \{w_{Ag}(a_j) : a_j \in A\}$.

Since we are concerned with the opinions of only one agent (the persuadee), we shall use the simplified notation w_j for $w_{Ag}(a_j)$.

Proof of Lemma 1: Let $X = \{w_1, ...w_n\} \subseteq W$ be the set of weights associated with the attributes examined. If $\mathcal{T}_1(X)$ holds, the algorithm terminates in at most n steps, by Definition 9.

Conversely if O_1 is determined in $n_1 \leq n$ steps after considering $X_1 \subseteq X$ then $\mathcal{T}_1(X_1)$ holds.

Subtracting $\sum_{w_j \in X \setminus X_1} w_j$ from both sides of $\mathcal{T}_1(X_1)$ gives

[Inequality 1]: $\sum_{w_j \in X_1} \tau_{1j} w_j - \sum_{w_j \in X_1} \tau_{2j} w_j - \sum_{w_j \in X \setminus X_1} w_j > \sum_{w_j \in W \setminus X} w_j,$

since $(W \setminus X_1) \setminus (X \setminus X_1) = W \setminus X$.

[1] We would like to thank Michael Bench-Capon for his insights regarding the proof strategy used in this section.

We aim to deduce that $\mathcal{T}_1(X)$ holds.

LHS of $\mathcal{T}_1(X) = \sum_{w_j \in X} \tau_{1j} w_j - \sum_{w_j \in X} \tau_{2j} w_j$

$\geq \sum_{w_j \in X_1} \tau_{1j} w_j - (\sum_{w_j \in X_1} \tau_{2j} w_j + \sum_{w_j \in X \setminus X_1} \tau_{2j} w_j)$

$\geq \sum_{w_j \in X_1} \tau_{1j} w_j - \sum_{w_j \in X_1} \tau_{2j} w_j - \sum_{w_j \in X \setminus X_1} w_j$ since $\tau_{2j} \leq 1$

$> \sum_{w_j \in W \setminus X} w_j$ (by Inequality 1). This is the RHS of $\mathcal{T}_1(X)$, so $\mathcal{T}_1(X)$ holds.

Hence the algorithm returns O_1 in no more than n steps (by discovering the status of some or all elements of A) if and only if $\mathcal{T}_1(X)$ holds. Similarly for O_2. Hence the probability $P(X)$ is independent of the order in which the elements of X are considered. □

Lemma 2: Let X, Y be n-element subsets of W which differ only in one element: $X = (X \cap Y) \cup \{w_x\}$ and $Y = (X \cap Y) \cup \{w_y\}$ with $w_x > w_y$. Then $P(X) \geq P(Y)$ where $P(X), P(Y)$ are as defined in Lemma 1.

Proof of Lemma 2: Let $X = \{w_1, ...w_{n-1}, w_x\}$, $Y = \{w_1, ...w_{n-1}, w_y\}$ where $w_x > w_y$. By Lemma 1, $P(X)$ and $P(Y)$ are well defined. We write TT, TF, FT, FF for the 4 possible values of $\langle \tau_{1j}, \tau_{2j} \rangle$. For each set X, Y, there are 4^n possible such assignments of truth values, all equally likely by our hypothesis. For example, when $n = 3$, one assignment is $\langle TT, FT, TT \rangle$, indicating the first and third attributes are true for both O_1 and O_2, but the second is true only for O_2. If the algorithm terminates for r out of the 4^n assignments, the probability of termination in at most n steps is $r/4^n$.

Suppose $\mathcal{T}_1(Y)$ holds for a particular assignment: we will show that it follows that $\mathcal{T}_1(X)$ holds for that assignment.

$\sum_{w_j \in X} \tau_{1j} w_j - \sum_{w_j \in X} \tau_{2j} w_j = \sum_{w_j \in Y} (\tau_{1j} - \tau_{2j}) w_j + (\tau_{1j} - \tau_{2j})(w_x - w_y)$

$\geq \sum_{w_j \in Y} (\tau_{1j} - \tau_{2j}) w_j - (w_x - w_y)$ since $\tau_{1j} - \tau_{2j} \in \{-1, 0, 1\}$ and $w_x > w_y$.

$> \left(\sum_{w_j \in W \setminus Y} w_j \right) - (w_x + w_y)$ by $\mathcal{T}_1(Y) = \sum_{w_j \in W \setminus X} w_j$ so that $\mathcal{T}_1(X)$ holds.

Hence the number r_X of assignments for which the algorithm terminates in not more than n steps is at least r_Y.

$P(X) = r_X/4^n$ and $P(Y) = r_Y/4^n$ so that $P(X) \geq P(Y)$ as required. □.

Proof of Theorem 1: Suppose there exists a strategy $\mathcal{R}$ better than $\mathcal{S}$. Then there exists $n \in \mathbb{N}$ such that $\mathcal{R}$ is more likely to terminate in at most n steps than $\mathcal{S}$.

Let $X_{\mathcal{S}} = \{w_1, w_2, ..., w_n\}$, be the set of the largest n weights and let $X_{\mathcal{R}}$ be the set of n weights used by $\mathcal{R}$.

We construct a chain of n-element sets $X_{\mathcal{S}}$, $X_1, X_2, ...X_k, X_{\mathcal{R}}$ so that each set X and its successor Y satisfy the conditions of Lemma 2. We retain elements of $X \cap Y$ and replace others in turn, substituting the largest in Y for the largest in X and continuing in descending order. For example, if X = {9,8,6,5} and Y = {8,4,3,1}, the intermediate sets are {4,8,6,5}, {4,8,3,5}.

Since $X_{\mathcal{S}}$ contains the largest weights, $w_x > w_y$ is satisfied for each pair in the chain. By Lemma 2, for each pair $\langle X, Y\rangle$, $P(X) \geq P(Y)$. Hence $P(X_{\mathcal{S}}) \geq P(X_{\mathcal{R}})$, a contradiction. So no such better strategy exists, and we conclude that $\mathcal{S}$ is optimal. □.

Suppose we relax the assumption that the persuadee knows nothing about the options in the initial situation, and instead starts with a set of initial (perhaps unverified) beliefs that lead to a preference for one of the options, but where there still remain attributes whose value for the options is unknown. This was the case for Bert in the Burger World example above. Corollary 1 demonstrates that $\mathcal{S}$ is an optimal strategy in this case also.

Corollary 1: If the utilities of O_1 and O_2 are initially unequal, $\mathcal{S}$ remains an optimal strategy.

Proof of Corollary 1: Let the current utilities be U_1, U_2 with $U_1 > U_2$ and $W = \{w_1.., w_m\}$ be the set of weights.

Suppose, for contradiction, that a better strategy $\mathcal{R}$ than $\mathcal{S}$ exists, expected to terminate in $n(\mathcal{R})$ steps with $n(\mathcal{R}) < n(\mathcal{S})$. Let L be a number greater than any weight in W.

Suppose there were additional attributes b_1, b_2 with weights $L + U_1, L + U_2$. Now initiate a dialogue with weights $W \cup \{L + U_1, L + U_2\}$ and the initial utilities zero for both options.

Apply the following strategy: Step 1: Inquire about b_1. Step 2: Inquire about b_2. After Step 2: if Step 1 assigns TF and Step 2 assigns FT, (difference between utilities is $U_1 - U_2$), continue as for $\mathcal{R}$, otherwise continue in descending order. For this strategy the expected number of steps is 2 + 1/4$n(\mathcal{R})$ + 3/4$n(\mathcal{S})$. The expected number of steps for the descending order strategy is 2 + $n(\mathcal{S})$ which is greater, contradicting Theorem 1. So no such strategy $\mathcal{R}$ exists. □

Next consider partial persuasion, where nothing improving the current estimated utility of the preferred option can be learned, since the persuader will choose to remain silent. Now all inquiry will take place concerning the option advocated by the persuader, values for the persuadee's preferred option being supplied by current beliefs and expectations. That $\mathcal{S}$ remains an optimal strategy in the case is established by Theorem 2.

Theorem 2: If τ_{1j} is known in advance for all attributes a_j, $\mathcal{S}$ is an optimal strategy for inquiring about the τ_{2j}.

Sketch of Proof of Theorem 2: In this case U_1 is constant throughout. After examining attributes with weights in $X \subseteq W$, U_2 = $\sum_{w_j \in X} \tau_{2j} w_j$.

The algorithm terminates when **either** $U_2 > U_1$
or $U_1 - U_2 > \sum_{w_j \in W \setminus X} w_j = \sum_{w_j \in W} w_j - \sum_{w_j \in X} w_j$.

U_2 increases monotonically.

The expected value of U_2 is $\frac{1}{2}\sum_{w_j \in X} w_j$ so the algorithm is expected to terminate when

either $\frac{1}{2}\sum_{w_j \in X} w_j > U_1$ **or** $\frac{1}{2}\sum_{w_j \in X} w_j > \sum_{w_j \in W} w_j - U_1$.

The RHS of each inequality is constant, so the best strategy is to maximise the LHS at each step, that is, to choose the largest remaining weight. But this is strategy $\mathcal{S}$ as in Theorem 1. □

Finally we consider the case where the persuadee, agent B, has initial beliefs about attributes relating to both options, but has varying degrees of confidence in these beliefs, and wishes to confirm them during the dialogue. Theorem 3 shows $\mathcal{S}$ to be an optimal strategy on the assumption that the currently preferred option is at least as likely as the alternative proposed by the persuader to satisfy the criteria valued by the persuadee. This will be the case where the persuadee is quite sure that its preferred option satisfies some desirable attributes, but has only tenuous beliefs about the other option.

Theorem 3: $\mathcal{S}$ remains an optimal strategy if the probabilities p_{ij} are not equal, provided that $p_{1j} \geq p_{2j}$ for each attribute j, where $p_{ij} = P(\tau_{ij} = 1)$.

Sketch of Proof of Theorem 3: After examining attributes with weights in $X \subseteq W$, (expected value of U_i) = $\sum_{w_j \in X} p_{ij} w_j$. So the algorithm is expected to terminate when

$$\sum_{w_j \in X} (p_{1j} - p_{2j}) w_j > \sum_{w_j \in W} w_j - \sum_{w_j \in X} w_j.$$

Arguing as before, an optimal strategy is, at each step, to choose from the remaining weights so as to maximise $w_j(p_{1j} - p_{2j} + 1)$. □

6 Concluding Remarks

We have identified the distinctive features of a sub-type of persuasion dialogue where one agent is trying to convince another that its currently preferred option is not as good as some other possibility known to the persuading agent. We have also drawn attention to the pragmatic meanings of utterances in these persuasion dialogues, as revealed by considering what the utterances conversationally imply, and have used these pragmatic meanings to develop a protocol and strategy for agent persuasion dialogues, which we have shown to be an optimal strategy in a range of representative scenarios for these persuasion dialogues. In future work we intend to consider dialogues with three or more participants, and dialogues which attempt to change the preferences of the participants.

References

1. Bench-Capon, T.: Agreeing to differ: Modelling persuasive dialogue between parties without a consensus about values. Informal Logic 22(32), 231–245 (2002)
2. Black, E., Hunter, A.: An inquiry dialogue system. Autonomous Agents and Multi-Agent Systems 19(2), 173–209 (2009)
3. García, A.J., Simari, G.R.: Defeasible logic programming: An argumentative approach. Theory and Practice of Logic Programming 4(1-2), 95–138 (2004)
4. Grice, H.: Logic and conversation. In: Cole, P., Morgan, J. (eds.) Syntax and Semantics, vol. 3, pp. 43–58. Academic Press, New York (1975)
5. McBurney, P., Hitchcock, D., Parsons, S.: The eightfold way of deliberation dialogue. International Journal of Intelligent Systems 22(1), 95–132 (2007)
6. Prakken, H.: Formal systems for persuasion dialogue. Knowledge Engineering Review 21(2), 163–188 (2006)
7. Searle, J.R.: Rationality in Action. The MIT Press, Cambridge (2003)
8. Walton, D.: Dialectical relevance in persuasion dialogue. Informal Logic 19(2-3), 119–135 (1999)
9. Walton, D., Krabbe, E.: Commitment in dialogue: Basic concepts of interpersonal reasoning. State University of New York Press, Albany (1995)

The Genoa Artificial Power-Exchange

Silvano Cincotti and Giulia Gallo

DOGE.I-CINEF, University of Genoa, Via Opera Pia 15, 16145 Genoa, Italy
{silvano.cincotti,giulia.gallo}@unige.it
www.cinef.it

Abstract. The paper presents the Genoa Artificial Power Exchange, an agent-based framework for modeling and simulating power exchanges implemented in MATLAB. GAPEX allows creation of artificial power exchanges reproducing exact market clearing procedures of the most important European power-exchanges. In this paper we present results from a simulation performed on the Italian PEX where we have reproduced the Locational Marginal Price Algorithm based on the Italian high-voltage transmission network with its zonal subdivisions and we considered the Gencos in direct correspondence with the real ones. An enhanced version of the Roth-Erev algorithm is presented so to be able to consider the presence of affine total cost functions for the Gencos which results in payoff either positive, negative and null. A close agreement with historical real market data during both peak- and off-peak load hours of prices reproduced by GAPEX confirm its direct applicability to model and to simulate power exchanges.

Keywords: Agent-based computational economics, Electricity markets, Reinforcement learning, Multi-agent systems.

1 Introduction

In the last decade, large efforts have been dedicated in developing theoretical and computational approaches for modeling deregulated electricity markets. Several papers have appeared in the agent-based computational economics (hereafter ACE) literature on wholesale electricity markets and ACE has become a reference paradigm for researchers working on electricity market topics (see as reference examples [17], [3], [4], [1], [7], and [22]). Generally speaking, these papers adopt a computer-based modeling approach for studying the electricity markets as result of the interactions between heterogenous market participants. In particular, the AMES model (Agent-based Modeling of Electricity Systems, [22]) comprised a two-settlement system consisting of a day-ahead market and a real-time one which are both cleared by means of Locational Marginal Pricing. [21] presented a model that consists of three sequential oligopolistic energy markets representing a wholesale gas market, a wholesale electricity market and a retail electricity market. [25] simulated two markets that are cleared sequentially: a day-ahead electricity market and a market for balancing power. [6] developed a wholesale electricity market model similar to the Australian National Electricity Market. Detailed reviews on agent-based models applied to wholesale electricity markets can be found in [26] and [12]. In this paper, we present the Genoa Artificial Power Exchange (GAPEX), an agent-based framework for modeling and simulating electricity markets. In particular,

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 348–363, 2013.

the general GAPEX framework is presented that allows us to generalize the models and to overcome some limitations and simplifications that characterized preliminary version of the framework ([7], [13] and [19]). In this paper, attention is devoted to model design and developing within GAPEX. This has direct implication on the features of the intelligent agents (i.e. Gencos) as well as on the mechanism of the power exchange. In particular, in order to properly model the decision process of the economic agents, an enhanced version of the classical Roth-Erev reinforcement learning algorithm [20] is described so to apply reinforcement learning in case of negative payoffs. Furthermore, due to its complex high-voltage transmission network, the Italian power exchange (IPEX) is taken as case of study. Results point out that GAPEX is an adequate framework to model and to simulate power exchanges. In particular, the agent-based model of the Italian Electricity Market is able to replicate market historical results during both peak- and off-peak load hours as well as to give insights on Genco behaviors. Moreover the proposed enhanced version of the classical Roth-Erev reinforcement learning algorithm points out effective learning properties with respect to existent variants in the literature.

The structure of the paper is as follows. In the next Section, the computational design and architecture of the GAPEX framework is presented. In Section 3 the Italian Electricity Day-Ahead Market agent-based model is described. In Section 4 the Enhanced Roth-Erev reinforcement learning algorithm is presented and studied. In Section 5 we present main results of the agent-based model of the Italian Power exchange, while Section 6 summarizes main results and remarks.

2 GAPEX Framework Overview

GAPEX is an agent-based framework developed in MATLAB that is suitable for studying the dynamic performances of many electricity markets. The simulator is implemented using OOP programming capabilities of MATLAB, which allows one to define classes using a Java/C++ like syntax, thus creating a flexible and extensible ABM framework which can run local simulation and also exploits the Parallel Computing Toolbox provided with MATLAB. Detailed computational models of the power techno-socio economic systems can be realistically simulated by means of the agent-based modeling (hereafter ABM) approach. Agents can range from entities with no cognitive function (e.g., transmission grids) to sophisticated decision makers capable of communication and learning (e.g., electricity traders). According to this research paradigm, we designed and implemented a versatile software framework for studying electricity markets. Indeed, the philosophy of the project and the modularity of its implementation provide a valuable computational framework for easily implementing other critical infrastructure systems relevant to energy markets, e.g., a natural gas market. In order to properly address the agent behaviors in different economic environments, we have used a multi-agent learning (MAL) approach so to define appropriate algorithms able to implement sophisticated decision-making rules. This represents one of the standards in the ACE literature and some common features characterize the learning models. The framework is composed by three main classes:

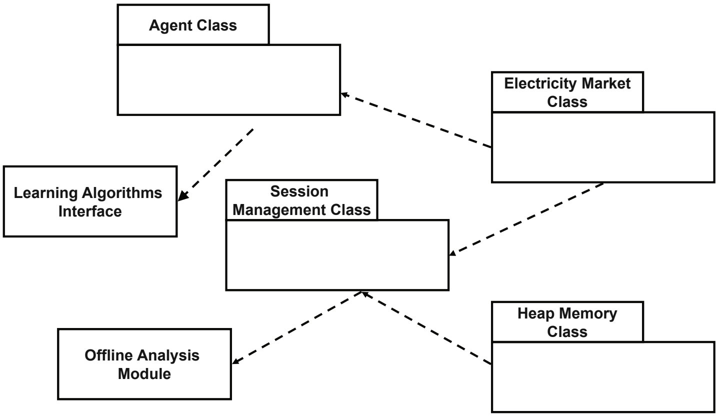


Fig. 1. GAPEX Class Architecture

- a heap class;
- a statistical off-line analysis module;
- several algorithms and market mechanisms libraries.

Figure 1 shows the GAPEX class architecture. The Agent class is an abstract class which is extended by all agents present in GAPEX Framework. It is worth noting that the Agent class is directly extended in order to define any new types of Electricity Market Agents (e.g., Wholesalers, Energy Management Divisions, etc). As concerning the learning algorithms, they are modeled as interfaces implemented by Gencos. Current version of GAPEX is characterized by a library of the main solutions for learning algorithms proposed in the literature (e.g., Roth-Erev algorithm, Q-Learning algorithm, Marimon-McGrattan algorithm, EWA learning and GiGa WoLF algorithm). In particular, these algorithms have been extended so to consider reward both positive, negative and null, and the features of the enhanced Roth-Erev algorithm are discussed in Section 4. The Electricity Market class allows one to define the market clearing algorithms and it is based on the Agent class. Currently, the GAPEX allows one to simulate the Italian Day Ahead Market, the EEX spot market linking DCOPFJ Package [22] and the Spanish Day Ahead Market. It is worth nothing that all these algorithms are interfaces as well. The Session class has a two-fold purposes. On the one hand, it acts as a clearing house and allows one to run several iterations of a particular simulation and to call the statistical off-line module at the end of the simulation. On the other hand, it stores all market and agent information, thus acting as a repository for all data related to energy prices and quantities both at market and at agent level (e.g. choices, propensities, etc). This feature is of crucial importance for economics application as it allows the GAPEX framework to be used as an artificial world where computational experiments can be performed. Indeed, such computational experiments are mandatory so to evaluate reproducibility of stylized facts as well as statistical properties of the self-adaptive complex system under investigation (see [2]). Moveover, in order to model the clearing house feature and characteristics, the mechanism of Heap memory access has been simulated and recreated into a MATLAB class. Thus, at the end of every simulation , the Clearing House recall the Offline Statistical Module which carry out statistical analysis as well as visualization of the computational experiment results. Finally, it is worth remarking that GAPEX allows direct generalization, as it is possible to create different types of agents, thus allowing the design of extremely realistic agent-based models.

3 Agent-Based Modeling of the Italian Electricity Day-Ahead Market

As discussed in previous Section, GAPEX is designed as a powerful and extensible agent-based framework for electricity market modeling and simulation. Current version of GAPEX allows one to simulate different power exchange protocols, but due to its complex structure, in this paper attention is dedicated to the Italian power exchange. It is worth remarking that a power exchange strongly differs from a stock market from both structural and behavioral point of view. From the former, the power exchange mechanism is a uniform double auction whereas the stock market one is continuous time limit order book. Furthermore, energy is not a storable good (i.e., *buy&hold* strategy are not even possible) whose consumption is contemporary to the production and is characterized by strong seasonality (i.e., daily, weekly and yearly). Moreover, from the latter, the electricity sector is characterized by strong oligopoly (i.e., a limited and basically time-invariant number of market traders) that repeat the same game on a daily base. Theoretically speaking, such economic system seems perfect for an analytical solution based on game theory, but the dimension of the game is so high that it practically impossible to study equilibria by means of traditional game theory. Despite a first glance on analytical solutions, all these elements lead to an economic system that can be effectively studied by means of a computational approach based on learning agents, thus motivating the development of GAPEX framework for the implementation of the model of the wholesale Italian Electricity Market. Making use of preliminary versions of GAPEX, [7] described and implemented an agent-based model of power exchange with a uniform price auction mechanism and a learning mechanism for the Gencos. Moreover, [13] provided the first version of the Genoa Artificial Power Exchange and compared the discriminatory and the uniform price auction mechanism with heterogenous agents. Finally, [19] firstly attempted to create an agent-based model of the Italian Electricity Market, with a reduced transmission network grid and a simplified description of GenCos. It is worth remarking that version presented and discussed in this paper of both the GAPEX and the agent-based model of the Italian electricity day-ahead market are characterized by significant extensions. Firstly, agent-based model incorporates now the exact procedure employed by the *Gestore Mercati Energetici S.p.A.* (hereafter GME) [9] thus overcoming the limitation of previously adopted formulation that resulted a constrained ill posed optimization procedure. Furthermore, the cognitive agents in the GAPEX make use of the Enhanced Roth-Erev reinforcement learning algorithm (presented and discussed in Section 4), developed so to take into account payoffs of any sign. There are crucial features that allowed us to calculate the energy prices based on scenarios that correctly emulate real power plants, real transmission limits and real bids. In this Section we present the agent-based model of the Italian Electricity

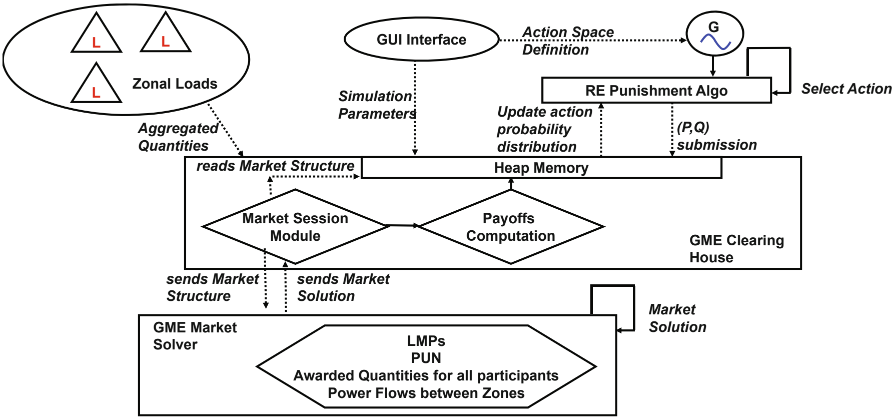


Fig. 2. ABM IPEX simulation flow-diagram

Day-Ahead Market (hereafter ABM IPEX Model). The Italian power exchange (IPEX) started on 1st April 2004 and is currently administrated by the Gestore Mercati Energetici S.p.A., the Italian market operator. IPEX market structure is characterized by several subsequent market sessions for both trading energy and managing critical services (e.g., reserves and real-time balancing). These are the Day-Ahead Market session - DAM, (Mercato del Giorno Prima - MGP), the Adjustment Market sessions and the Ancillary Services Market. The most important (i.e., liquid) session is the Day-Ahead Market which is organized as a non-discriminatory double-auction market where approximately 60 percent of national production is traded. The main feature of the Italian Day-Ahead Market is related to the complex high-voltage transmission network and results in a zonal splitting with both locational and national energy prices. The ABM IPEX simulation flow-diagram (i.e., static representation of the objects and their interactions) is shown in Figure 2. It is worth remarking that the ABM IPEX Model consists of three main building blocks, i.e.:

- the agent-based representation of the Italian Electricity Market and the clearing mechanism regarding the Day-Ahead Market;
- the representation of the Italian Electricity Network;
- the agent-based representation of traders in the Italian Electricity Market, i.e. Gencos.

These building blocks are discussed in the following sub-sections.

3.1 Day-Ahead Market Model

GAPEX simulates Gencos bidding strategies through a daily market session in the Italian Electricity DAM. The exact market clearing procedure performed by Italian Market Operator has been implemented (see [8] for a detailed discussion). Furthermore, the following agents are currently represented in the model:

- Gencos: They are the economic actors at the supply side of the electricity market. They submit supply bids to the GME Market Operator and (after the market

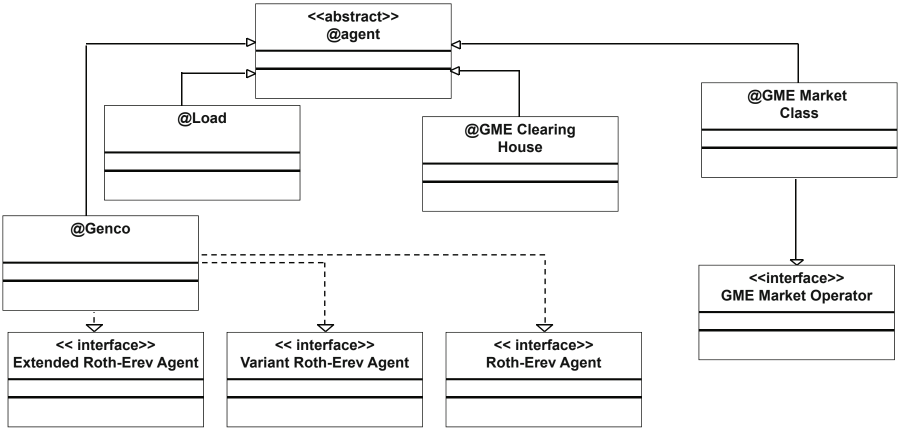


Fig. 3. UML class diagram of agents in ABM IPEX

clearing procedure) they access the GME clearing house in order to retrieve market results and to update their strategic decisions. They extend GAPEX Agent class;

- Loads: They are aggregations of zonal loads and represent the demand side of the electricity market as inelastic;
- GME Market Operator: It clears the market and sends information on awarded prices and quantities to the GME clearing house. It extends GAPEX Electricity Market class;
- GME Clearing House: It computes all payoffs for the Gencos, updates their market accounts and stores all market information. It extends GAPEX Session class.

It is worth remarking that the aim of the proposed model is to represent and to study the strategic behavior of Gencos in the power exchange. Accordingly, the Gencos are characterized by sophisticated decision process (i.e., the Enhanced Roth-Erev reinforcement learning algorithm presented and discussed in Section 4)) that accounts for the effect of a repeated game. Furthermore, according to the hypothesis of a competitive electricity market, the Gencos communicate directly only with the GME Market Operator and GME Clearing House so to account that every Genco is only aware of its own strategies and payoffs. Finally, all the other agents in the model are passive entities and they are not endowed with any cognitive capability. Figure 3 shows the UML class diagram for the agents modeled in the ABM IPEX:

At each iteration step, each i^{th} generator ($i = 1, 2, ..., N$) submits to the DAM a bidding curve shown in Figure 4. The curve is described by the triple of P_i ([€/MWh]), Q_i^- ([MWh]),Q_i^+ ([MWh]), i.e., the bidding price, the minimum and the maximum production power for i^{th} generator, respectively. After receiving all generators' bids, the DAM clears the market by performing a social welfare maximization subject to the constraints on the zonal energy balance (Kirchhoff's laws) and on inter-zonal transmission limits (see [8] for details). The objective function takes into account only the supply side of the market as the demand is assumed to be price-inelastic. The zonal splitting clearing mechanism (i.e., DC optimal power flow procedure) allows one to determine both the unit commitment for each generator and the Locational Marginal Price (LMP) for each

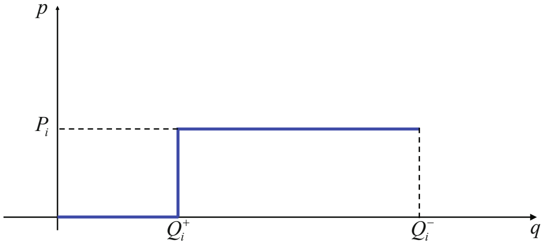


Fig. 4. Reference bidding curve for a Genco

zone. To this aim, a graph representation of the transmission grid (that defines the area with relevant transmission constraints) is provided as input to the GAPEX (see Section 3.2). However, with respect to classical literature on power systems, the Italian market introduces two modifications. Firstly, sellers are paid at the zonal prices, i.e., Location Marginal Price (LMP), whereas buyers pay a unique national price (Prezzo Unico Nazionale - PUN) common for the whole market and computed as a weighted average of the zonal prices with respect to the zonal loads. Secondly, transmission power-flow constraints differ according to the flow direction which results in doubling the number of constraints related to the inter-zonal transmission limits. According to the specific features of the Italian market, the results of the power exchange auction consist of a set of the active powers Q_i^* and of a set of Locational Accepted Marginal Prices LMP_k for each zone $k \in \{1, 2, ..., K\}$.

3.2 Transmission Grid Model

The market clearing procedure described in Section 3.1 requires the definition of a transmission network. The grid structure adopted in this paper is shown in Figure **??** and reproduces the exact zonal market structure and the relative maximum transmission capacities between neighboring zones of the Italian grid model as indicated by Terna S.p.A. the Italian transmission system operator. The relevant areas of the network correspond to physical geographic areas (e.g. Northern Italy, Sicily, Sardinia, etc.) in which loads and generators, virtual production areas (i.e. foreign neighboring countries) or limited production areas (e.g. Priolo Gargallo) are present. It is worth remarking that each zone is represented as a bus to whom generators and loads are connected. Furthermore, the arches linking the zone represent the transmission connections and account for the constraints in transmissions for the power flow. Finally, transmission power-flow constraints differ according to the flow direction, e.g., power flowing from Central-South to Central-North is subject to a transmission limit that is different from the one relates to the power flowing from Central-North to Central-South. A detailed discussion of the Italian transmission grid can be found in [23].

3.3 Genco Model

The supply side of the market is composed by Gencos submitting bids for each of their power plants. In this paper attention is focussed on thermal power plants strategic

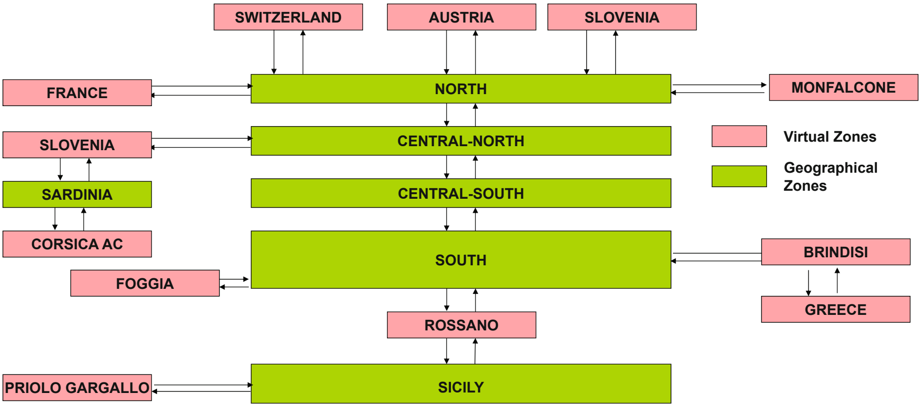


Fig. 5. The Italian grid model

behavior, as the remaining national production (i.e., hydro, geothermal, solar, wind) and imported production can be generally modeled as bids at zero price [16]. A set of thermal power plants consisting of N = 175 generating units is considered. These comprise five different technologies (i.e., Coal-Fired (CF), Oil-Fired (OF), Combined Cycle (CC), Turbogas (TG) and Repower (RP)) and in the model a learning agent is associated to each generating unit. The constant marginal costs of the i^{th} generator is assumed to be given by:

$$MC_i = \pi_i \quad [\text{€/MWh}] \tag{1}$$

The coefficients π_i has been selected using an econometric analysis on real historical bids. The total cost function of i^{th} generator is assumed to be given by:

$$TC_i(Q_i) = a_i \cdot Q_i + b_i \quad [\text{€/h}] \tag{2}$$

The coefficients a_i ([€/MWh]) and b_i ([€/h]) are assumed constants. a_i depends mainly on the class of efficiency and on the technology of the power plant, whereas b_i (which is specific for each power plant) accounts for investment and other quasi-fixed costs that must be recovered and that are not negligible for capital intensive industry such as the electricity one. As a consequence, the coefficients a_i have been evaluated on the basis of $MC_i(Q_i)$ with fuel costs, technology and efficiency as exogenous variables, whereas the coefficients b_i have been determined by the literature on technological business cases [15]. Stated the cost functions of the Gencos, it is necessary to define the decision process that drives the bidding strategy. In this respect, we assume that the bidding price P_i of the i^{th} generator (see Section 3.1) is a mark-up μ_i applied to the marginal cost MC_i in equation 1, i.e.,

$$P_i = (1 + \mu_i) \cdot MC_i \tag{3}$$

As a consequence, the decision variable of the i^{th} generator is the mark-up μ_i and the learning process should individuate a profitable value for μ_i as results of the interaction (through the energy market) with the other Gencos. In particular, the profit $R_i(h)$

depends on the market clearing at hour h. Assuming that the i^{th} Genco belongs to zone k, $R_i(h)$ is given by

$$R_i(h) = LMP_k(h) \cdot Q_i^*(h) - TC_i(Q_i^*(h)) \ \ [\text{€/h}] \tag{4}$$

where TC_i is i^{th} Genco total-cost, $LMP_k(h)$ is the Location Marginal Price of zone k at hour h and $Q_i^*(h)$ is the awarded quantity to the i^{th} Genco at hour h. Finally, it is worth remarking that the marginal cost is the reference parameter for the bids (see equation 3), whereas the total costs are crucial in order to evaluate the real profitability of the bids (see equation 4).

4 Enhanced Roth-Erev Reinforcement Learning Algorithm

Electricity markets are characterized by inherent complexity and repeated games that requires adequate modeling of strategic behavior of traders. This is usually achieved by endowing the Gencos with learning capability. The literature on agent-based electricity market models points out three major kind of learning algorithms: *zero-intelligence* algorithms [10],[11], *reinforcement and belief-based* models [5] and *evolutionary* approach [18]. In this paper, the strategic agent behavior is modeled by means of a reinforcement learning approach. It is worth remarking that the solutions proposed in the literature generally account for positive and null payoffs (e.g., [18] represented a first modification of the original work proposed by Roth and Erev [20] so to account for null payoffs). Unfortunately, this is a severe limitation in order to determine profitable strategy for economic agents in real a economic context. Indeed, the presence of fixed-costs in the cost function (see equation 2) together with market awarded quantity $Q_i^*(h) \geq 0$ for the i^{th} Genco at hour h leads to payoffs that are either positive, negative or null. This opens a question for a reinforcement learning approach that is able to cope with payoffs of any sign and to this aim we have developed an enhanced version of the Roth and Erev algorithm that is able to cope with both positive, negative and null payoffs. The original Roth and Erev learning model (hereafter referred to as RE algorithm) considers three psychological aspects of human learning:

- the power law of practice, i.e., learning curves are initially steep and tend to progressively flatten out;
- the recency (or forgetting) effect, i.e., players recent experience plays a larger role than past experience in determining his behavior;
- the experimentation effect, i.e., not only experimented strategy but also similar strategies are reinforced.

For each strategy $a_j \in \mathcal{A}_j$ $(i = 1, .., M)$, at every round t, propensities $S_{j,t-1}(a_j)$ are updated according to:

$$S_{j,t}(a_j) = (1 - r) \cdot S_{j,t-1}(a_j) + E_{j,t}(a_j) \tag{5}$$

where $r \in [0, 1]$ is the recency parameters which contributes to decrease exponentially the effect of past results. The second term of equation 5 is called the experimentation function and is given by:

$$E_{j,t}(a_j) = \begin{cases} \mathit{\Pi}_{j,t}(\hat{a}_j) \cdot (1-e) & a_j = \hat{a}_j \\ \mathit{\Pi}_{j,t}(\hat{a}_j) \cdot \frac{e}{\mathbf{M}-1} & a_j \neq \hat{a}_j \end{cases} \tag{6}$$

where $e \in [0,1]$ is the experimentation parameter which assigns different weights between the played strategy and the non-played strategies and $\mathit{\Pi}_{j,t}(\hat{a}_j)$ is the reward obtained by playing strategy $(\hat{a}_j)$ at round t. Propensities are then normalized so to determine the probability for the strategy selection policy $\pi_{j,t+1}(a_j)$ for the next auction round as:

$$\pi_{j,t+1}(a_j) = \frac{S_{j,t}(a_j)}{\sum_{a_j} S_{j,t}(a_j)} \tag{7}$$

The modified Roth and Erev learning model (hereafter referred to as MRE algorithm) by [18] proposed a solution for the case of zero payoffs by modifying the experimentation function in equation 6 according to:

$$E_{j,t}(a_j) = \begin{cases} \mathit{\Pi}_{j,t}(\hat{a}_j) \cdot (1-e) & a_j = \hat{a}_j \\ S_{j,t-1}(a_j) \cdot \frac{e}{\mathbf{M}-1} & a_j \neq \hat{a}_j \end{cases} \tag{8}$$

It is worth remarking that MRE and RE are identical for a positive reward $\mathit{\Pi}_{j,t}(\hat{a}_j)$, whereas for null payoff MRE introduces an implicit *premium* for non-played strategies with respect to the ineffective (i.e. with negative $\mathit{\Pi}_{j,t}(\hat{a}_j)$) played strategy. MRE represents a first but not final extension of the Roth and Erev algorithm as neither MRE algorithm nor the later VRE algorithm proposed by [22] are able to cope with negative payoffs. In order to overcome such limitation of the Roth-Erev algortihm, we propose to extend the MRE algorithm by enhancing the experimentation mechanism for non-played strategies according to:

$$E_{j,t}(a_j) = \begin{cases} G[\mathit{\Pi}_{j,t}(\hat{a}_j)] \cdot (1-e) & a_j = \hat{a}_j \\ F[\mathit{\Pi}_{j,t}(\hat{a}_j)] \cdot S_{j,t-1}(a_j) \cdot \frac{e}{\mathbf{M}-1} & a_j \neq \hat{a}_j \end{cases} \tag{9}$$

where

$$G[x] = \begin{cases} -\gamma \cdot \tanh(x) & x \geq 0 \\ 0 & x \leq 0 \end{cases} \tag{10}$$

and

$$F[x] = \begin{cases} \alpha \cdot \tanh(x) + 1 & x \leq 0 \\ 1 & x \geq 0 \end{cases} \tag{11}$$

Figure 6 shows functions $G[...]$ and $F[...]$. It is worth noting that the proposed enhanced version represents an extension of the MRE. In particular, in the case of negative payoff, the experimentation function for the played strategies is calculated as in MRE proposed by [17] for the case of null payoffs, whereas the experimentation function of the non-played strategies is enhanced by a larger amplification the more negative is the payoff $\mathit{\Pi}_{j,t}(\hat{a}_j)$. This leads to an Enhanced Roth and Erev algorithm (hereafter referred to as ERE algorithm). In the simulations discussed hereafter, we have adopted the values of 0.12 and 0.20 for the parameters e and r, respectively. Moreover, the value of 3.0 and

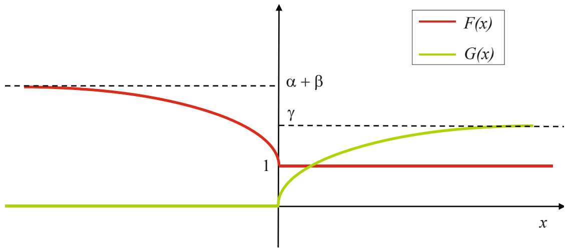


Fig. 6. Functions involved in ERE

10.0 have been chosen for the parameters α and γ, respectively. It is worth noting that the values for e, r, α, ad γ have been chosen so to guarantee stability of the difference equations involved in the learning process (i.e., equations 5 and 9). In order to understand effectiveness of the proposed Enhanced Roth and Erev algorithm and the interrelation between learning convergence and economic results, we firstly studied the behavior and the convergence of the learning in the power exchange model. We have assumed the initial (i.e., at $t = 0$) propensities $S_{j,t}(a_j)$ in equation 5 to be uniformly distributed among the possible strategies in the strategy space. Furthermore, as discussed in Section 3.3, the strategy space is related to the mark-up variables. In all computational experiments discussed hereafter we have considered a uniformly spaced grid for μ_i in the range [0.8, 2.3] with step 0.05. This results in a set of 31 possible strategies for each of the N = 175 generators. Stated this simulation context, the evolution of the strategy probabilities pointed out three groups of agents:

- those whose bids are lower than clearing prices and are always accepted by the market. We denote them as price-takers agents and are characterized by a convergence of the strategy probabilities;
- those whose bids are higher than clearing prices and are always rejected by the market. We denote them as out-of-the-market agents and are generally characterized by randomly chosen strategies, as they do not participate to the market price formation and accordingly receive always negative payoffs;
- those whose bids are able to set the Locational Marginal Price. We denote them as price-maker agents and are characterized by the faster convergence time in the learning process.

Figure 7 shows an example of reference convergence time-path. For the sake of representativeness, the strategy characterized by the largest final probability (i.e., the action most willing to be played) of three reference Gencos is considered and their probabilities plotted as function of the simulation iterations. Figure 7 points out that both price-taker and price-maker are characterized by a learning process that select the preferred action strategy (i.e., the one whose probability converge to 1). Conversely, it is worth noting that only some of the out-of-the-market agents are characterized by a convergence of the strategy probabilities. Indeed, those agents whose bids are slightly higher than the LMP tend to converge even if their bids are always rejected by the market. This can be interpreted as a result of an almost complete exploration process of their strategy spaces that allows them to conclude that the strategies played by the near competitors (i.e., the price maker agents) were characterized by a bidding price lower enough to keep them out of the market. In this exploration process, they are characterized by the slower convergence time, thus corroborating such conclusion.

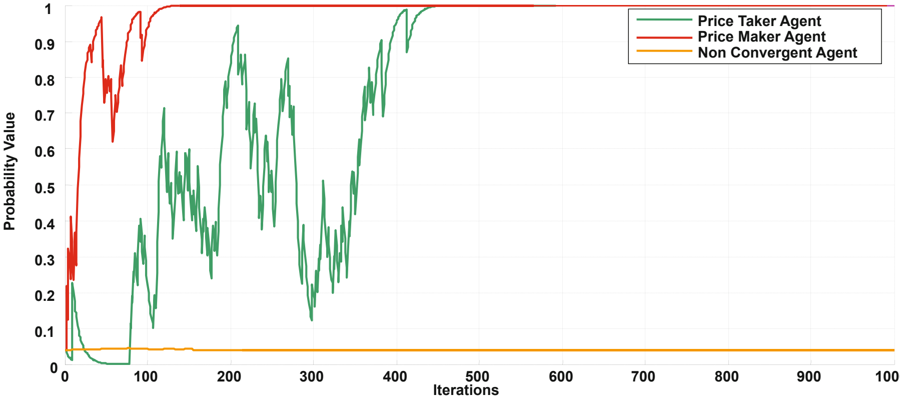


Fig. 7. Convergence time-path for the different groups of interacting agents in ABM IPEX

5 Computational Experiments

Learning algorithms and agent-based models should stick to empirical criteria in order to demonstrate that they are able to reproduce reality. In particular, at the micro-level, learning algorithms should converge toward a price during the experiments, whereas, at the macro-level, practitioners should be able to observe stylized facts and economic emergent behaviors. Completed the learning convergence (see Section 4), we focussed our attention to a set of computational experiments in order to understand the ability of the framework to reproduce the emergent properties shown by the IPEX DAM at macro-level. Firstly, we have chosen a reference power exchange setting (i.e., Gencos and loads). In this respect, the scenario has been based on a real off-peak hour (i.e., hour 5 AM of Wednesday 16th December 2009) as during off-peak hour competition among producers is generally limited and thus limiting the impact on the level of prices. For the reference power exchange setting, we have performed 100 computational experiments with different random seeds in order to analyze the ensemble results of the same repeated game. Both agent convergence and system convergence have been observed. While the former has been discussed in Section 4 and used as a validation proof of the enhanced Roth-Erev learning algorithm, we now concentrate on the system convergence. This type of convergence (or its lack) can be defined with respect to the convergence of the PUN time path (i.e. the clearing price converge to a value after a specific time which depends only from the participating agents). Indeed, the PUN is a weighted-average of the Locational Marginal Prices by means of the inelastic loads and an adequate representative of the market clearing and its convergence a good proxy that the system has reached an equilibrium. In this context, we searched for the ”best performing” economic-learning algorithm, i.e. the selection of the algorithm should have lead both to learning convergence and to economic meanings. Due to the proportional update mechanism of the strategy selection probability (see Section 4), this is a real system convergence and not a fictitious one induced by a cooling parameter. We also observe that at the aggregate the learning process achieves an equilibrium that corresponds to a local optimum rather than to a global one, as the PUN and LMP dependent both on profits (i.e. payoffs) and on strategy spaces. Similar fictitious results have been already discussed for the Roth-Erev algorithm in a simplified agent-based electricity model (see [14]) as well as for Q-Learning (see [24]). Furthermore, in the case of VRE learning algorithm the shape of the curve suggests that although the probabilities of strategy spaces of the agents have been updated during the simulation, prices at the beginning of the simulation are the same as at the end. This directly points out that agents have not learnt any preferred strategy (i.e. there is no convergence) and leads to a ”random noise shape” of the prices, as discussed in [14]. It is worth remarking that these results further point out effectiveness of the proposed Enhanced Roth and Erev algorithm (with the respect to the other state-of-the-art version proposed by the literature) and its direct applicability to economic and financial context characterized by positive, null and negative rewards. Finally, the complete 24 hours PUNs of Wednesday 16th December 2009 have been simulated. Again, we have performed 100 computational experiments (each with a length of 5,000 steps) with different random seeds in order to analyze the ensemble results of the same repeated game. It is worth remarking, that the energy market is characterized by a strong seasonality (i.e., daily, weekly, and yearly), therefore the strategic behavior of Gencos can be properly studied on a daily base. Figure 9 compares the GAPEX simulated PUNs to the GME real PUNs. Figure 9 points out that the simulated results are in good agreement throughout the whole 24 hours. Indeed, most of the GME real PUNs fall within the 95 percent (i.e., $2^*\sigma$) confidence band evaluated over the 100 computational experiment whereas the outliers are however quite close to the limit of the 95 percent confidence band. This further states the quality and importance of the proposed methodology which is mostly able to replicate the aggregate results by means of the strategic interactions of the Gencos rather than of a black-box forecast. Understanding the origin of the market results is a crucial element from an economic point of view as it allows us to determine the drivers and model of the power exchange. Every policy measure, antitrust action and market design requires a clear understanding of these elements in order to be effective. Furthermore, it is worth noting that in the case of the computational experiments, the generation universe is kept fixed with cost functions unchanged for the whole 24 hours. This has been assumed in order to evaluate the ability of the learning algorithm for selecting the most profitable strategy in different condition of demands. However, such condition is not present in the real GME market sessions as the generation plants are characterized by outages. The absence of outages in the computational experiments can explain the small difference between GAPEX simulated PUNs to the GME real PUNs and it is worth noting that including outages in the GAPEX is easy and direct. However, such an interesting scenario for computer science results of limited interest from an economics perspective. Indeed, it is characterized by such a large $ex - ante$ information (the exact information of the hourly participation of the Gencos to the power auction) that it results practically irrelevant and for this reason it has not been considered. Finally, the good agreement between the GAPEX simulated PUNs and the GME real PUNs achieved by the strategic computational experiments remarks the importance of including fixed-costs in the decision-making process of Gencos. Indeed, results point out a strong relationship between fixed-costs and profits that the Enhanced Roth-Erev algorithm was able to incorporate thus improving realism of the model.

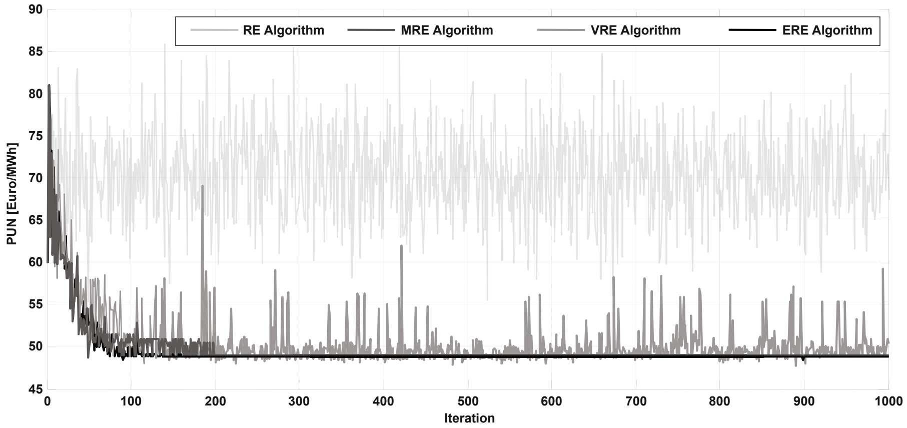


Fig. 8. Convergence time-path of PUN in ABM IPEX using reinforcement learning algorithms

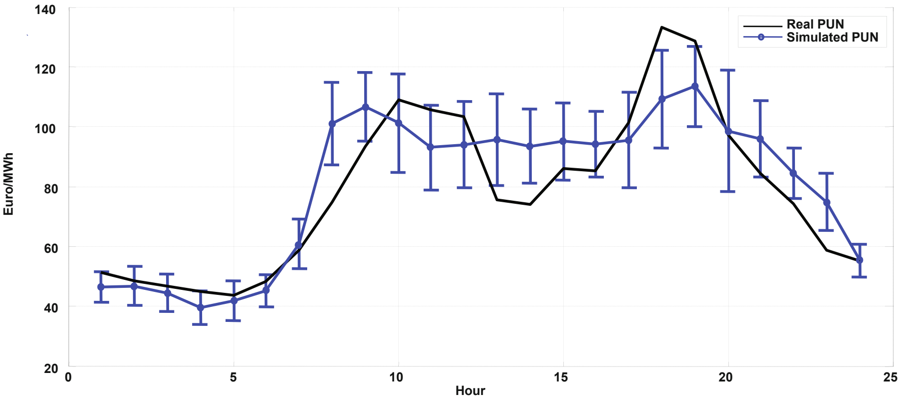


Fig. 9. 24 hours GAPEX simulated PUNs vs. GME real PUNs. The bars denote 95% confidence interval.

6 Conclusions

In this paper, an agent-based electricity market framework has been presented. The framework has been implemented in MATLAB using the OOP paradigm and it allows creation of artificial power exchanges characterized by real market mechanisms and by economic agents with learning capabilities. In order to overcome limitations in the sign of payoff typical of reinforcement learning algorithms proposed in the literature, an enhanced version of the Roth-Erev algorithm (i.e., which takes into account positive, null and negative payoffs) has been presented and discussed. Furthermore, due to its complex high-voltage transmission network, the Italian power exchange (IPEX) has been taken as case of study. This resulted in replicating the exact market clearing procedure and considering generation plants in direct correspondence with the real ones. Results on the convergence of the enhanced Roth-Erev learning algorithm pointed out effectiveness of the proposed algorithm. In particular, the evolution of the strategy probabilities pointed out different groups of agents characterized by different convergence rates that strongly depend on the role of the agent in the market. This fact confirms the direct applicability of the proposed Enhanced Roth-Erev learining algorithm for economic and financial applications. Moreover, computational experiments of the ABM IPEX model performed within the GAPEX pointed out a close agreement with historical data during both peak- and off-peak load hours, confirming the direct applicability of the GAPEX to model and to simulate power exchanges in particular for what-if analysis and market design.

Acknowledgements. E. Guerci and M.A. Rastegar collaborated to the design and the development of the GAPEX framework. This work has been partially supported by the University of Genoa, by the Italian Ministry of Education, University and Research (MUR) under grant PRIN 2007, by the European Social Fund (ESF) and by Regione Liguria, Italy.

References

1. Bagnall, A., Smith, G.: A multi-agent model of the UK market in electricity generation. IEEE Transactions on Evolutionary Computation 9(5), 522–536 (2005)
2. Ball, P.: The earth simulator. New Scientist 2784, 48–51 (2010)
3. Bower, J., Bunn, D.W.: Experimental analysis of the efficiency of uniform-price versus discriminatory auctions in the England and wales electricity market. Journal of Economic Dynamics & Control 25, 561–592 (2001)
4. Bunn, D.W., Oliveira, F.: Agent-based simulation: an application to the new electricity trading arrangements of England and wales. IEEE Transactions on Evolutionary Computation 5(5), 493–503 (2001)
5. Camerer, C., Ho, T.: Experience-weighted attraction learning in normal-form games. Econometrica 67, 827–874 (1999)
6. Cau, T.D.H., Anderson, E.J.: A co-evolutionary approach to modelling the behaviour of participants in competitive electricity markets. In: IEEE Power Engineering Society Summer Meeting, vol. 3, pp. 1534–1540 (2002)
7. Cincotti, S., Guerci, E., Raberto, M.: Agent-based simulation of power exchange with heterogeneous production companies. Computing in Economics and Finance 2005, Society for Computational Economics 334 (2005)
8. GME: Official web site (2010), `http://www.mercatoelettrico.org/En/Default.aspx`
9. GME: Uppo auction module user manual, appendix a - market splitting auction algorithm. Tech. rep., GME (2010), `http://www.mercatoelettrico.org/En/MenuBiblioteca/Documenti/20100429MarketSplitting.pdf`

10. Gode, D.D.K., Sunder, S.: Allocative efficiency of markets with zero intelligence traders. Market as a partial substitute for individual rationality. J. Polit. Econ. 101(1), 119–137 (1993)
11. Gode, D.D.K., Sunder, S.: Double auction dynamics: structural effects of non-binding price controls. Journal of Economic Dynamics and Control 28(9), 1707–1731 (2004)
12. Guerci, E., Rastegar, M., Cincotti, S.: Agent-based modeling and simulation of competitive wholesale electricity markets. Handbook of Power Systems 3(2), 241–286 (2010)
13. Guerci, E., Ivaldi, S., Raberto, M., Cincotti, S.: Learning oligopolistic competition in electricity auctions. Computational Intelligence 23(2), 197–220 (2007)
14. Jing, Z., Ngan, H., Wang, Y., Zhang, Y., Wang, J.: Study on the convergence property of roth-erev learning model in electricity market simulation. In: 8th International Conference on Advances in Power System Control, Operation and Management (APSCOM 2009), pp. 1–5 (November 2009)
15. Kirschen, D.S., Strbac, G.: Fundamentals of Power System Economics. Wiley (2004)
16. Migliavacca, G.: Srems: a short-medium run electricity market simulator based on game theory and incorporating network constraints. In: Power Tech, 2007 IEEE Lausanne, Switzerland, pp. 813–818 (July 2007)
17. Nicolaisen, J., Petrov, V., Tesfatsion, L.: Market power and efficiency in a computational electricity market with discriminatory double-auction pricing. IEEE Transactions on Evolutionary Computation 5(5), 504–523 (2001)
18. Nicolaisen, J., Smith, M., Petrov, V., Tesfatsion, L.: Concentration and capacity effects on electricity market power. In: Proceedings of the 2000 Congress on Evolutionary Computation, La Jolla, USA, vol. 2, pp. 1041–1047 (2000)
19. Rastegar, M., Guerci, E., Cincotti, S.: Agent-based model of the Italian wholesale electricity market. In: Proceedings of the 6th International Conference on the European Energy Market, EEM 2009 (2009)
20. Roth, A.E., Erev, I.: Learning in extensive form games: Experimental data and simple dynamic models in the intermediate term. Games Econ. Behav. 8(1), 164–212 (1995)
21. Ruperez Micola, A., Banal Estaol, A., Bunn, D.W.: Incentives and coordination in vertically related energy markets. Journal of Economic Behavior and Organization 67, 381–393 (2008)
22. Sun, J., Tesfatsion, L.: Dynamic testing of wholesale power market designs: An open-source agent-based framework. Comput. Econ. 30, 291–327 (2007)
23. TERNA S.p.A.: Individuazione della rete rilevante - italian version only. Tech. rep., TERNA S.p.A. (2008)
24. Watkins, C., Dayan, P.: Q-learning. Machine Learning 8(3-4), 279–292 (1992)
25. Weidlich, A., Veit, D.J.: Bidding in interrelated day-ahead electricity markets: Insights from an agent-based simulation model. In: Proceedings of the 29th IAEE International Conference, Potsdam (2006)
26. Weidlich, A., Veit, D.J.: A critical survey of agent-based wholesale electricity market models. Energy Economics 30, 1728–1759 (2008)

Manipulation of Weighted Voting Games via Annexation and Merging

Ramoni O. Lasisi and Vicki H. Allan

Department of Computer Science, Utah State University, UT 84322-4205, Logan, U.S.A.
ramoni.lasisi@aggiemail.usu.edu, vicki.allan@usu.edu

Abstract. We conduct an experimental study of the effects of *manipulations* (i.e., dishonest behaviors) including those of manipulation by *annexation* and *merging* in weighted voting games. These manipulations involve an agent or agents misrepresenting their identities in anticipation of gaining more power at the expense of other agents in a game. Using the well-known Shapley-Shubik and Banzhaf power indices, we first show that manipulators need to do only a polynomial amount of work to find a much improved power gain, and then present two enumeration-based pseudopolynomial algorithms that manipulators can use. Furthermore, we provide a careful investigation of heuristics for annexation which provide huge savings in computational efforts over the enumeration-based method. The benefits achievable by manipulating agents using these heuristics also compare with those of the enumeration-based method which serves as upper bound.

Keywords: Agents, Weighted voting games, Annexation, Merging, Power indices.

1 Introduction

False-name manipulation in *weighted voting games* (WVGs), which involves an agent or some agents misrepresenting their identities in anticipation of power increase, has been identified as a problem. The menace can take different forms. With manipulation by *annexation*, an agent, termed, an *annexer*, takes over the voting weights of some agents in a game. Power is not shared with the annexed agents. Forming an *alliance* or manipulation by *merging* involves voluntary merging of weights by two or more agents to form a single bloc [2,13,18]. Merged agents expect to be compensated with their share of the power gained by the bloc. The agents whose voting weights are taken over or merged into a bloc are referred to as *assimilated* agents. The only difference between merging and annexation is that, in merging, the assimilated agents must be compensated for their participation by sharing of the power, while in annexation, only the annexer is compensated for the participation of the group. Annexed agents are assumed to either voluntary forfeit their weight or be compensated on a one-time basis that is not related to power. Thus, power increase is much easier to achieve with annexation.

WVGs are classic cooperative games which provide compact representation for coalition formation models in human societies and multiagent systems. Each agent in a WVG has an associated weight. A subset of agents whose total weight is at least the value of

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 364–378, 2013.

a specified *quota* is called a *winning coalition*. The weights of agents in a game correspond to resources or skills available to the agents, while the quota is the amount of resources or skills required for a task to be accomplished. For example, in *search and rescue*, robotic agents put their resources (i.e., weights) together in large natural disaster environments to reach the necessary levels (i.e., quota) to save life and property.

The relative power of each agent reflects its significance in the elicitation of a winning coalition. A widely accepted method for measuring such relative power in WVGs uses *power indices*. Two prominent power indices for measuring power are the *Shapley-Shubik* [23] and the *Banzhaf* [6] power indices. WVGs can be viewed as a form of competition among agents to share the available *fixed* power whose total value is always assumed to be 1. Agents may thus resort to manipulation by annexation or merging to improve their influence in anticipation of gaining more power. With the possibility of manipulation, it becomes difficult to establish or maintain trust, and more importantly, it becomes difficult to assure fairness in such games.

This paper continues the work of [2,13,18,19] on annexation and merging in WVGs. We first extend the framework of [18] on susceptibility of power indices to manipulation by annexation and merging in WVGs to consider a *much improved* power gain for manipulators. Then, we propose and evaluate heuristics that manipulating agents may employ to engage in such manipulations using the two power indices. Consider a WVG of n agents. The simulation of [18] is based on a *random* approach where some agents, say $k < n$, in the game are randomly selected to be assimilated in annexation or to form a voluntary bloc of manipulators in merging. This simple random approach shows that, on average, annexation can be effective for manipulators using the two power indices to compute agents' power. These results also show that merging only has a minor effect on the power gained for manipulators using the Shapley-Shubik index, while it is typically non-beneficial (i.e., no power is gained) for manipulators using the Banzhaf index.

Restricting the number of agents in the blocs of assimilation or merging as employed in the simple random simulation of [18], we show that manipulators need to do only a *polynomial* amount of work to find a much improved power gain over the random approach during manipulation. Given that the problem of computing the Shapley-Shubik and Banzhaf indices of agents is already NP-hard [21,22], *pseudopolynomial* or approximation algorithms [7,9,21] are available to compute agents' power. We then present two pseudopolynomial-time enumeration algorithms that manipulators may use to find a much improved power gain. We empirically evaluate our enumeration approach for manipulation by annexation and merging in WVGs. Our method is shown to achieve significant improvement in benefits over previous work for manipulating agents in several numerical experiments. Thus, unlike the simple random simulation of [18] where merging has little or no benefits for manipulators using the two power indices, results from our experiments suggest that manipulation via merging can be *highly* effective.

The remainder of the paper is organized as follows. Section 2 provides some preliminaries. In Section 3, we provide visual illustrations of manipulation in WVGs to give some insights into why it is difficult to predict how to merge. We present our pseudopolynomial-time manipulation algorithms for annexation and merging in Section 4. Section 5 presents results of evaluation of the manipulation algorithms. In Sections

6, we present heuristics for annexation in WVGs. Section 7 discusses related work and we conclude in Section 8.

2 Preliminaries

2.1 Weighted Voting Games

Let $I = \{1, \cdots, n\}$ be a set of n agents and the corresponding positive weights of the agents be $\mathbf{w} = \{w_1, \cdots, w_n\}$. Let a coalition $S \subseteq I$ be a non-empty subset of agents. A WVG G with *quota* q involving agents I is represented as $G = [w_1, \cdots, w_n; q]$. We assume that $w_1 \leq w_2 \leq \ldots \leq w_n$. Denote by $w(S)$, the weight of a coalition, S, derived as the summation of the weights of agents in S, i.e., $w(S) = \sum_{j \in S} w_j$. A coalition, S, wins in game G if $w(S) \geq q$, otherwise it loses. WVGs belong to the class of *simple voting games*. In simple voting games, each coalition, S, has an associated function $v : S \rightarrow \{0, 1\}$. The value 1 implies a win for S and 0 implies a loss. So, $v(S) = 1$ if $w(S) \geq q$ and 0 otherwise.

2.2 Power Indices

We provide brief descriptions of the two power indices we use in computing agents' power in WVGs. For further discussion, we refer the reader to [12,17].

Shapley-Shubik Power Index

The Shapley-Shubik index quantifies the marginal contribution of an agent to the *grand coalition* (i.e., a coalition of all the agents). Each permutation (or ordering) of the agents is considered. We term an agent *pivotal* in a permutation if the agents preceding it do not form a winning coalition, but by including this agent, a winning coalition is formed. Shapley-Shubik index assigns power to each agent based on the proportion of times it is pivotal in all permutations. We specify the computation of the index using notation of [2]. Denote by π, a permutation of the agents, so $\pi : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$, and by Π the set of all possible permutations. Denote by $S_\pi(i)$ the predecessors of agent i in π, i.e., $S_\pi(i) = \{j : \pi(j) < \pi(i)\}$. The Shapley-Shubik index, $\varphi_i(G)$, for each agent i in a WVG G is

$$\varphi_i(G) = \frac{1}{n!} \sum_{\pi \in \Pi} [v(S_\pi(i) \cup \{i\}) - v(S_\pi(i))]. \tag{1}$$

Banzhaf Power Index

An agent $i \in S$ is referred to as being *critical* in a winning coalition, S, if $w(S) \geq q$ and $w(S \backslash \{i\}) < q$. The Banzhaf power index computation for an agent i is the proportion of times i is critical compared to the total number of times any agent in the game is critical. The Banzhaf index, $\beta_i(G)$, for each agent i in a WVG G is given by

$$\beta_i(G) = \frac{\eta_i(G)}{\sum_{j \in I} \eta_j(G)} \tag{2}$$

where $\eta_i(G)$ is the number of coalitions for which agent i is critical in G.

2.3 Annexation and Merging in Weighted Voting Games

Felsenthal and Machover [13] consider a real life example of annexation where a shareholder buys the voting shares of some other shareholders in a firm in order to use them for her own interest. Clearly, this action allows the manipulator to possess more shares and makes it easier for her to affect the outcomes of decisions in the firm. Yokoo et al., [25] have also considered false-name manipulation in open anonymous environments which they refer to as *collusion*. Like in merging, collusion involves many agents acting as a single agent. They have shown that the manipulation can be difficult to detect in such environments. Thus, the increase use of online systems (such as trading systems and peer-to-peer networks, where WVGs are also applicable) means that annexation or merging remains an important challenge that calls for attention. We now provide a formal definition of manipulation by annexation and merging in WVGs.

Let G be a WVG. Let Φ be either of Shapley-Shubik or Banzhaf power index. We denote the power of an agent i in G by $\Phi_i(G)$. Also, consider a coalition $C \subseteq I$, we denote by $\&C$ a bloc of assimilated voters formed by agents in C. We say that a power index Φ is *susceptible* to manipulation whenever a WVG G is *altered* by an agent i (in the case of annexation or by some agents in the case of merging) and such that there exists a new game G' where $\Phi_i(G') > \Phi_i(G)$. In other words, Φ is susceptible to manipulation when the power of the agent in the altered game is more than its power in the original game.

Definition 1. *Manipulation by Annexation.*

Let agent i alter game G by annexing a coalition C ($i \notin C$ assimilates the agents in C to form a bloc $\&(C \cup \{i\})$). We say that Φ is susceptible to manipulation via annexation if there exists a new game G' such that $\Phi_{\&(C\cup\{i\})}(G') > \Phi_i(G)$; the annexation is termed *advantageous*. The *factor of increment* by which the annexer gains is given by $\frac{\Phi_{\&(C\cup\{i\})}(G')}{\Phi_i(G)}$. If $\Phi_{\&(C\cup\{i\})}(G') < \Phi_i(G)$, then the annexation is *disadvantageous*.

We provide an example to illustrate annexation in WVG. The annexer and assimilated agents are all shown in bold.

Example 1. Annexation in Weighted Voting Game.

Let $G = [\mathbf{12}, 16, \mathbf{18}, \mathbf{19}, 23, 26, 43, 46, 50; 195]$ be a WVG. The Banzhaf power index of agent 1 with weight 12 is $\beta_1(G) = 0.026$. Suppose the agent annexes agents 3 and 4 with weights 18 and 19 respectively. An assimilated bloc of weight 49 is formed in the new game $G' = [16, 23, 26, 43, 46, \mathbf{49}, 50; 195]$. The new Banzhaf power index of the annexer $\beta_6(G') = 0.177 > \beta_1(G)$. The agent gains from the annexation and increases its power index by a factor of $\frac{0.177}{0.026} = 6.81$.

Definition 2. *Manipulation by Merging.*

Let a manipulators' coalition, S, alter G by merging into a bloc $\&S$. We say that Φ is susceptible to manipulation via merging if there exists a new game G' such that $\Phi_{\&S}(G') > \sum_{j\in S} \Phi_j(G)$; the merging is termed *advantageous*. The factor of increment by which the manipulators gain is given by $\frac{\Phi_{\&S}(G')}{\sum_{j\in S}\Phi_j(G)}$. If $\Phi_{\&S}(G') < \sum_{j\in S} \Phi_j(G)$, then the merging is *disadvantageous*. The agents in a bloc formed by merging are assumed to be working cooperatively and have transferable utility. For the sake of simplicity in our analysis, we also refer to the factor of increment as power gain or benefit.

Example 2 illustrates manipulation by merging in a weighted voting game.

Example 2. Merging in Weighted Voting Game.

Let $G = [12, 16, 18, 19, 23, \mathbf{26}, \mathbf{33}, \mathbf{40}, \mathbf{45}; 155]$ be a WVG. The last four agents in the game are designated as would-be manipulators. The Banzhaf indices of these agents are: $\beta_6(G) = 0.116$, $\beta_7(G) = 0.142$, $\beta_8(G) = 0.174$, and $\beta_9(G) = 0.200$. So, $\sum_{j=6}^{9} \beta_j(G) = 0.632$. Suppose the agents decide to merge their weights. A bloc of weight 144 is formed in the new game $G' = [12, 16, 18, 19, 23, \mathbf{144}; 155]$. The Banzhaf power index of the bloc $\beta_6(G') = 0.861 > 0.632$. The manipulators gain from the merging and increase their power indices by a factor of $\frac{0.861}{0.632} = 1.36$.

3 Visual Description of Manipulation by Annexation and Merging in Weighted Voting Games

We provide visual description of manipulation by annexation and merging in WVGs to further explain the intricases of what goes on during manipulation. We use the Shapley-Shubik power index for this illustration. Consider a WVG of three agents denoted by the following patterns: Agent 1 (■), Agent 2 (□), and Agent 3 (▤). The weight of each agent in the game is indicated by the associated length of the pattern. A box in the pattern corresponds to a unit weight. Suppose all permutations (or ordering) of the three agents are given as shown in Figure 1 where we vary the values of the quota of the game from $q = 1$ to $q = 6$. The Shapley-Shubik indices of the three agents are also computed from the figure and shown in the associated table of the figure. These power indices for the agents in the game correspond to using various values of the quota for the same weights of the agents in the game.

Consider the case of manipulation by merging where Agent 1 and Agent 3 merge their weights to form a new agent, say Agent X. In this situation, Agent 1 and Agent 3 cease to exist since they have been assimilated by Agent X. Thus, we have only two agents (Agent X and Agent 2) in the altered WVG. Figure 2 shows the results of the merging between Agent 1 and Agent 3. Consider the cases when the quota of the game is 1 or 6, the power of the assimilated agents for Agent X from Figure 1 shows that Agent 1 and Agent 3 each has a power of $\frac{1}{3}$ for a total power of $\frac{2}{3}$. Whereas, the power of Agent X which assimilates these two agents in the two cases is each $\frac{1}{2} < \frac{2}{3}$. On the other hand, the power of the manipulators stays the same for the cases where the quota is either 2 or 5. Specifically, the sum of the powers of Agent 1 and Agent 3 is $\frac{1}{2}$ for these cases. This is also true of Agent X for these cases. Finally, for the cases where the quota of the game is either 3 of 4, the power of Agent X is 1 which is greater than $\frac{5}{6}$, the sum of the powers of Agent 1 and Agent 3 in the original game. Note the difficulty of predicting what will happen when manipulators engage in merging.

Now, suppose Agent 1 annexes Agent 3 instead of the merging between the two agents as described earlier, then Agent 3 ceases to exist as shown in Figure 3. In this case, Agent 1's power does not decrease. Felsenthal and Machover [12] have already

Permutation	q = 1	q = 2	q = 3	q = 4	q = 5	q = 6
1	■	■	■	□	□	▤
2	▤	■	■	■	□	□
3	▤	□	□	■	■	■
4	□	□	■	■	■	▤
5	□	□	▤	■	■	■
6	■	■	■	▤	□	□
Agent 1(■)	$\frac{2}{6}$	$\frac{3}{6}$	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{3}{6}$	$\frac{2}{6}$
Agent 2(□)	$\frac{2}{6}$	$\frac{3}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{3}{6}$	$\frac{2}{6}$
Agent 3(▤)	$\frac{2}{6}$	$\frac{0}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{0}{6}$	$\frac{2}{6}$

Fig. 1. Six permutations of three agents and corresponding power indices of the agents for various values of quota from $q = 1$ to $q = 6$

Permutation	q = 1	q = 2	q = 3	q = 4	q = 5	q = 6
1	■	■	■	■	▤	▤
2	▤	▤	■	■	■	■
Agent X(■)	$\frac{1}{2}$	$\frac{1}{2}$	1	1	$\frac{1}{2}$	$\frac{1}{2}$
Agent 2(▤)	$\frac{1}{2}$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{2}$

Fig. 2. Manipulation by merging between Agent 1 and Agent 3 (from Figure 1) to form a new Agent X. The power indices of Agent X and Agent 2 as computed by Shapley-Shubik index after the merging for various values of the quota is also shown.

shown that annexation is never disadvantageous for an annexer using the Shapley-Shubik power index. However, that is not the case when the annexer uses the Banzhaf power index. This they have dubbed the bloc paradox.

4 Manipulation Algorithms for Annexation and Merging

4.1 Overview

We propose an enumeration approach for manipulation by annexation and merging in WVGs using the two power indices to compute agents' power. To begin with, we recall that the problem of calculating the Shapley-Shubik and Banzhaf indices is NP-hard and admit pseudopolynomial algorithms using generating functions or dynamic

Permutation	q = 1	q = 2	q = 3	q = 4	q = 5	q = 6
1	■	■	■	■	□	□
2	□	□	■	■	■	■
Agent X(■)	$\frac{1}{2}$	$\frac{1}{2}$	1	1	$\frac{1}{2}$	$\frac{1}{2}$
Agent 2(□)	$\frac{1}{2}$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{2}$

Fig. 3. Manipulation by annexation where Agent 1 assimilates Agent 3 (from Figure 1). The new power indices of Agent 1 as computed by Shapley-Shubik index after the annexation for various values of the quota is also shown.

programming [7,9,21]. Given that the problem of computing the two indices is already NP-hard, and only pseudopolynomial or approximation algorithms are available to compute agents' power, it is reasonable that the manipulation algorithms we propose are also pseudopolynomial since we necessarily need to use these indices in computing agents' benefits during manipulation.

4.2 Manipulation Algorithm for Merging

The brute force approach to determine a coalition that yields the most improved benefit in merging in a WVG is to simply enumerate all the possible coalitions of agents in the game and then compute the benefit for each of these coalitions. We can then output the coalition with the highest value. Unfortunately, enumerating all the possible coalitions is exponential in the number of agents. Also, computing the power indices naively from their definitions means that we have two exponential time problems to solve. We provide an alternative approach.

Let procedure *PowerIndex*(G, i) be a pseudopolynomial algorithm for computing the power index of an agent i in a WVG G of n agents for Shapley-Shubik or Banzhaf index according to any of [7,9,21]. We first use *PowerIndex*(G, i) as a subroutine in the construction of a procedure, *GetMergeBenefit*(G, S). Procedure *GetMergeBenefit*(G, S) accepts a WVG G and a would-be manipulators' coalition, S. It first computes the sum of the individual power index of the assimilated agents in S using *PowerIndex*(G, i). Then, it alters G by replacing the sum of the weights of the assimilated agents in G with a single weight in a new game G' before computing the power of the bloc $\&S$ in G'. Finally, *GetMergeBenefit*(G, S) returns the factor of increment of the merged bloc $\&S$. Let $A(G)$ be the pseudopolynomial running time of *PowerIndex*(G, i). Now, since $|S| \leq |I| = n$, procedure *GetMergeBenefit*(G, S) takes at most $O(n \cdot A(G))$ time which is pseudopolynomial.

We now use *GetMergeBenefit*(G, S) to construct an algorithm that manipulators can use to determine a coalition that yields a good benefit in merging. We first argue that manipulators tend to prefer coalitions which are small in size because they are easier to form and less likely to be detected. Also, intra-coalition coordination, communication, and other overheads increase with coalition size. Thus, we suggest a limit on the size of the manipulators' coalitions since it is unrealistic and impractical that *all* agents in a WVG will belong to the manipulators' coalition. This reasoning is consistent with the assumptions of the previous work on annexation and merging [2,18] as well as coalition formation [24]. We note, however, that limiting the manipulators coalitions' size this way does not change the complexity class of the problem as finding the coalition that yields the most improved benefit remains NP-hard.

Consider a WVG of n agents. Suppose the manipulators' coalitions, S, have a limit, $k < n$, on the size of the members of the coalitions, i.e., S, are bounded as $2 \leq |S| \leq k$. In this case, the number of coalitions that the manipulators need to examine is at most $O(n^k)$ which is polynomial in n. Specifically, the total number of these coalitions is:

$$\binom{n}{2} + \binom{n}{3} + \cdots + \binom{n}{k} = \sum_{j=2}^{k} \binom{n}{j}. \tag{3}$$

So, we have

$$\begin{aligned}\sum_{j=2}^{k} \binom{n}{j} &= \sum_{j=2}^{k} \frac{n(n-1)\cdots(n-j+1)}{j!} \\ &\leq \sum_{j=2}^{k} \frac{n^j}{j!} \\ &\leq \sum_{j=2}^{k} \frac{n^j}{2^{j-1}} \\ &= \frac{n^2}{2^1} + \frac{n^3}{2^2} + \cdots + \frac{n^k}{2^{k-1}} = O(n^k).\end{aligned}$$

Running *GetMergeBenefit*(G, S) while updating the most[1] improved benefit found so far from each of these coalitions requires a total running time of $O(n^k \cdot A(G))$ which is pseudopolynomial, and thus becomes reasonable to compute.

4.3 Manipulation Algorithm for Annexation

Our pseudopolynomial manipulation algorithm for annexation provides a modification of the merging algorithm. We first replace *GetMergeBenefit*(G, S) with another procedure, *GetAnnexationBenefit*(G, i, S). *GetAnnexationBenefit*(G, i, S) accepts a WVG G, an annexer, i, and a coalition S to be assimilated by i. The procedure then returns the factor of increment of the assimilated bloc $\&(S \cup \{i\})$.

We use *GetAnnexationBenefit*(G, i, S) to construct an algorithm that the annexer can use to determine the coalition that yields the most improved benefit in annexation. The

[1] We refer to the most improved benefit among the $O(n^k)$ polynomial coalitions and not from the original 2^n coalitions since we have restricted each manipulators' coalition size to a constant $k < n$.

method of construction of the algorithm is the same as that of the manipulation algorithm for merging with the exception that we add the weight of an annexer i to the weight of each coalition S and compare the power index $\Phi_{\&(S\cup\{i\})}(G')$ of the assimilated bloc in a new game G' to the power index $\Phi_i(G)$ of the annexer in the original game G. The annexer examines a polynomial number of coalitions of the agents assuming a limit $k < n$ on the size of each coalition. Since the annexer belongs to a coalition it annexes, the total number of coalitions examined by the annexer is:

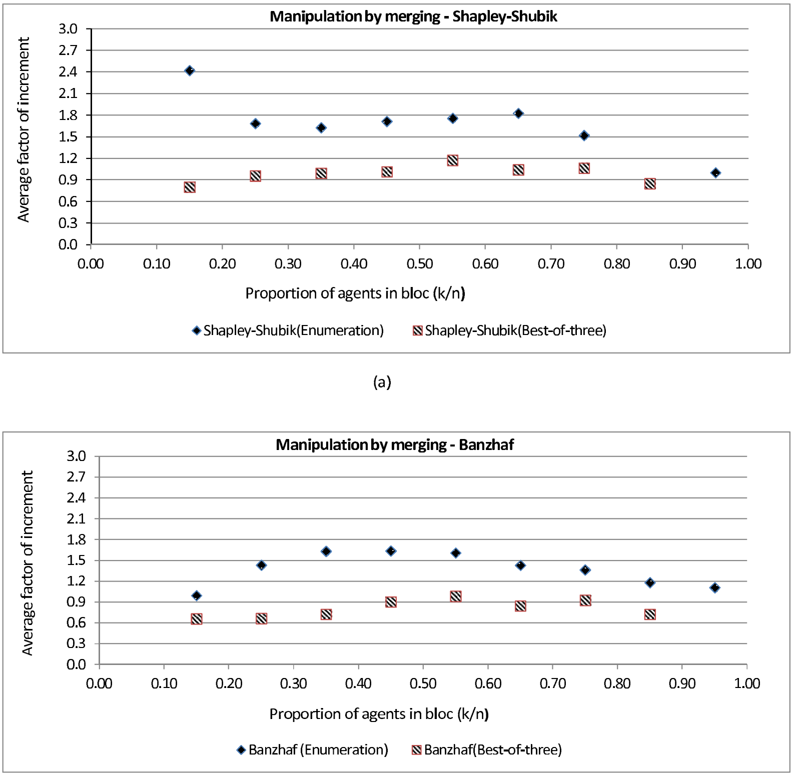
$$\binom{n-1}{1} + \cdots + \binom{n-1}{k-1} = \sum_{j=1}^{k-1} \binom{n-1}{j}. \tag{4}$$

Bounding this equation using similar approach as in Equation 3 shows that Equation 4 is $O(n^k)$. Thus, as before, the manipulation algorithm for annexation also runs in pseudopolynomial time, with a total running time of $O(n^k \cdot A(G))$.

5 Evaluation of the Manipulation Algorithms

We first argue that the simple random simulation of [18] for manipulation by annexation and merging is unintelligent. Thus, it is impractical that strategic agents would employ such method to engage in manipulation [19]. This is because simply guessing a particular coalition from among all the exponential possible coalitions provides a rare chance for strategic agents to benefit in both manipulation by annexation and merging. We note also that this chance of success by the manipulators decreases as the number of agents in a game becomes large. Now, since it is impractical to exhaustively consider all the exponential possible coalitions, we have presented two pseudopolynomial manipulation algorithms where we have restricted the sizes of coalitions to be considered by the manipulators to a constant k which is less than the number n of the agents in the game. Our idea is for the manipulators to sacrifice optimality for good merging or annexation. By doing so, the manipulation algorithms potentially bypass a lot of search. Although, the manipulation algorithms are incomplete, nonetheless, they consider more search space than the simple random approach, and hence, are guaranteed to find a much improved factor of increment than the simple random simulation of [18].

We perform experiments to confirm the above hypothesis. First, we make a simple modification to the random simulation of [18] which provides manipulators with higher average factor of increment. The modification involves the selection of the best factor of increment from three random choices (which we refer to as the *best-of-three* method). We compare results of our enumeration-based method with those of the best-of-three method. We randomly generate WVGs. The weights of agents in each game are chosen such that all weights are integers and drawn from a normal distribution, $N(\mu, \sigma^2)$, where μ and σ^2 are the mean and variance. We have used $\mu = 50$ and values of standard deviation σ from the set $\{5, 10, \ldots, 40\}$. When creating a new game, the quota, q, of the game is randomly generated such that $\frac{1}{2}w(I) < q \le w(I)$, where $w(I)$ is the sum of the weights of all agents in the game. The number of agents, n, in each of the original WVGs is chosen uniformly at random from the set $\{10, 11, \ldots, 20\}$ while the number of assimilated agents k is uniformly chosen at random from the set $\{5, 6, \ldots, 10\}$.



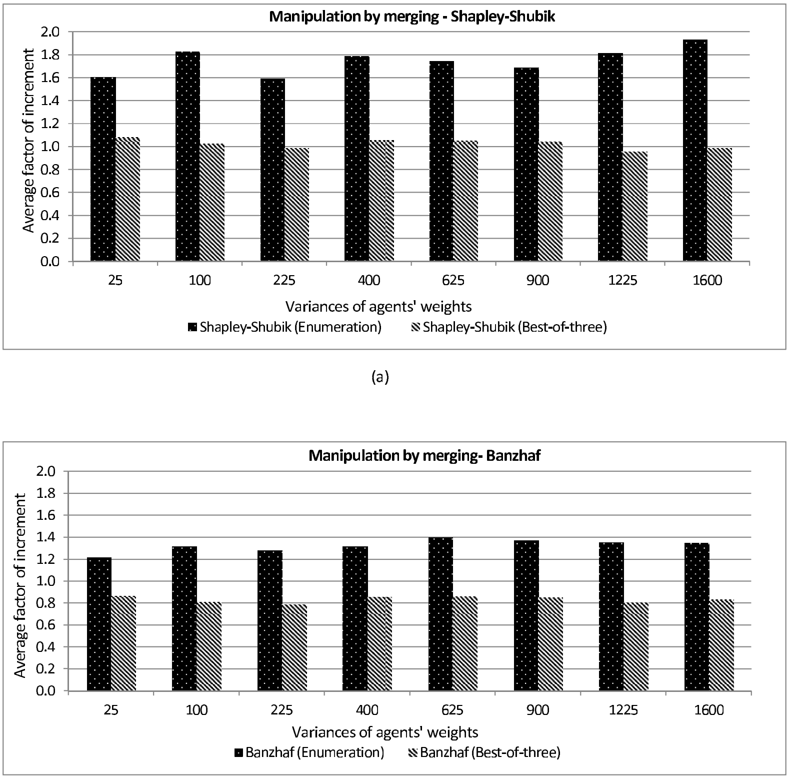
(a)

(b)

Fig. 4. The average factor of increment for merging for the enumeration and best-of-three methods using various values of size of bloc divided by the number of agents in the games

We have used a total of 500 original WVGs for each experiment and compute the average factor of increments over the entire set of games for the two power indices. The evaluation is carried out for the proportion of agents in manipulators' blocs and the variance of the weights of agents in the WVGs. We have implemented efficient exact polynomial methods of computing the Shapley-Shubik and Banzhaf indices for integer weights using generating functions [7,9] which can be exponential-time in complexity in the worst case.

We first present the results of the case of manipulation by merging and then provide discussion on the results of manipulation by annexation. Figure 4 shows the benefits from merging for both the best-of-three and enumeration-based methods using the two indices. The x-axes indicates the proportion of agents in the manipulators' bloc (i.e. $\frac{|S|}{|I|} = \frac{k}{n}$) whose factor of increment were reported while the y-axes are the average factor of increment achieved by manipulating agents in those coalitions. For the Shapley-Shubik power index, Figure $4(a)$ shows that manipulating agents achieved improved power using the enumeration approach than using the best-of-three method. There are cases where the manipulators achieved more than 2.4 times as much as the original power for the enumeration method, and in general the average factor of increment



(b)

Fig. 5. The average factor of increment for merging for the enumeration and best-of-three methods using different variances of agents' weights

is between 1.5 and 2.7 times the original power of the manipulators. Whereas, the best-of-three method has only minor effects for the manipulators as the average factors of increment in these tests are below 1.3. Similar trends between the enumeration and best-of-three methods are observed for the case of Banzhaf index too (See Figure $4(b)$). However, the average factor of increment is lower for the two methods using the Banzhaf index. On average, merging does not appear to significantly improve power using the best-of-three method for either the Shapley-Shubik or Banzhaf index, and in most cases is harmful for the agents. We conclude that since improvement in power over the best-of-three method can be achieved with only a polynomial amount of work, then, manipulators are more likely to seek a much improved power gain in merging using the enumeration-based approach.

Figure 5 provides further comparison of the enumeration and best-of-three methods for the two indices using different variances of agents' weights. It is clear from both Figures $5(a)$ and $5(b)$ that irrespective of the variances in the weights of agents in the games, the enumeration method is better in all cases. As before, the figures also show that on average, the best-of-three method does not appear to improve average factor of increment. Finally, for the case of manipulation by annexation and for both the Shapley-Shubik and Banzhaf indices, the impovement in power achieved by manipulating agents using the enumeration method is by far more than those achieved using the best-of-three method.

6 Heuristics for Annexation and Merging in WVGs

Unlike the manipulation algorithms for annexation and merging of Section 4 where we have n and k as the number of agents and the number of assimilated agents in a WVG, manipulating agents may not be interested in achieving the most improved power gain among the $O(n^k)$ polynomial coalitions described before. This is because the number of these coalitions may be large even for small values of n and k. Thus, we propose heuristics that agents may use for annexation in WVGs. Considering the basic requirements for a good heuristic as its ease of computation and to be as informative as possible [14], the heuristics we propose are designed to first avoid the enumeration approach of the manipulation algorithms in Section 4. Second, we avoid the unintelligent simple random approach of [18] to ensure that our heuristics provide good information for manipulating agents to make decisions on how to annex in WVGs.

6.1 Annexation Heuristics

We recall the definition of annexation in Section 2 and from [2,13], the power of the assimilated bloc in an altered WVG is compared to the power of the annexer in the original game. By this definition, intuition suggests that annexation should always be advantageous. This intuition is indeed true using the Shapley-Shubik index to compute agents' power. However, there exist situations where annexation is disadvantageous for the annexer using the Banzhaf index. See [2,3,13] for examples of WVGs where annexation is disadvantageous for the annexer using the Banzhaf index. This case where annexation results in power decrease for the annexer is refer to as the *bloc paradox* [13].

Again, recall from Equation 4 that an annexer needs to examine a polynomial number of assimilated coalitions of size at most $k - 1$ to find the most improved power gain among these coalitions whose sizes we have restricted to k. We note also that the assimilated coalitions with maximal weights among these coalitions are those of sizes $k - 1$. Thus, it is enough for a particular annexer to check only the assimilated coalitions of size exactly $k - 1$ in order for the annexer to find the coalition with the most improved benefit using the two power indices. This indeed is the case for the Shapley-Shubik index. For the case of Banzhaf index, we conduct a test to check the highest factor of increment among all the coalitions of $2,000$ different WVGs. We found that the coalitions with maximal weights (i.e., those having sizes of $k - 1$) yield the highest possible factor of increment in over 82% of the games. The remaining highest factor of increment are archieved by manipulators' blocs with lower number of agents in them. In none of these two situations do we experience the bloc paradox. This suggests that the bloc paradox for the Banzhaf power index may be a rare phenomenon in practice.

There are only $\binom{n-1}{k-1}$ such assimilated coalitions to be considered by this annexer. As seen, the amount of work carried out by the annexer is still polynomial, however, this heuristic requires smaller computational effort compared to that of the enumeration-based method and thus makes it more useful in practice. We can even do better using the idea in the following heuristic:

> **MaximalWeights Heuristic:**
> *Given*: A WVG of n agents with a distinguished annexer i.
> *Procedure*: Since agents' weights are given in ascending order, let agent i annex the last $k - 1$ agents from the remaining $n - 1$ agents in the game.

As before, let $A(G)$ be the pseudopolynomial running time of Shapley-Shubik or Banzhaf power index of an agent according to [7,9]. The running time of the *MaximalWeights* heuristic is $O(n + A(G))$. Here is a brief analysis. We just need to sum the weights of the last $k - 1$ agents from the remaining $n - 1$ agents to that of the annexer at a cost of $O(k)$. In the worst case, it takes $O(n)$ to sum the weights when $k = n$, i.e., the annexer is able to annex all the remaining $n - 1$ agents. Computing the power of the annexer in each of the original and altered games takes $O(A(G))$. Therefore, the total running time of the heuristic is $O(n + A(G))$. Now, apart from the fact that this heuristic appears to find the most improved gain, its running time is by far less than the $O(n^k \cdot A(G))$ running times of the manipulation algorithm for annexation of Subsection 4.3 and the above heuristic especially when k is large. Thus, the *MaximalWeights* heuristic is more useful in practice for the annexer.

7 Related Work

Weighted voting games and power indices are widely studied [8,12,17]. Prominent real-life situations where WVGs have found applications include the United Nations Security Council, the International Monetary Fund [1,20], the Council of Ministers, and the European Community [12]. The need to compensate agents from jointly derived payoff in WVGs has also necessitated the assignment of power to players. A widely accepted method for measuring power of agents in WVGs uses power indices. *Fairness* in the assignment of power to players in a game is also a concern of most of the power indices. The two most prominent and widely used power indices are Shapley-Shubik [23] and Banzhaf [6] power indices. Other power indices found in the literature include Deegan-Packel [10], Johnsoton [16], and Holler-Packel [15] power indices. Computing the Shapley-Shubik and Banzhaf power indices of players in WVGs is NP-hard [22]. The power indices of voters using any of Shapley-Shubik and Banzhaf power indices can be computed in pseudo-polynomial time using dynamic programming [21]. Efficient exact algorithms using generating functions [7,9] also exist for both the Shapley-Shubik and Banzhaf power indices for WVGs where the weights of all agents are restricted to integers. There are also approximation algorithms [5,11] for computing the Shapley-Shubik and Banzhaf power indices in WVGs.

Felsenthal and Machover [12,13] originally studied annexation and alliance (or merging) in WVGs. They consider when the blocs formed by annexation or merging are advantageous or disadvantageous. They show that using the Shapley-Shubik index, it is always advantageous for a player to annex some other players in the game. However, this is not true for Banzhaf power index. Furthermore, they show that merging can be advantageous or disadvantageous for the two power indices. In contrast to our work, they do not consider the extent to which the agents involved in annexation or merging may gain, which we study in this paper. Aziz et al. [2], have also considered the computational aspects of the problem of annexation and merging in WVGs. They show that determining if there exists a beneficial merge in a WVG is NP-hard using both the Shapley-Shubik and Banzhaf indices. The same is also true for determining the existence of beneficial annexation using the Banzhaf index.

8 Conclusions

We investigate the effects of manipulations (i.e., dishonest behaviors) by annexation and merging in weighted voting games. These manipulations involve an agent or agents misrepresenting their identities in anticipation of gaining more power in a game. We evaluate the effects of these manipulations using two prominent power indices, *Shapley-Shubik* and *Banzhaf* indices, to compute agents' power. We first provide visual illustrations of manipulation in weighted voting games to give some insights into why it is difficult to predict how to merge. We then show that manipulators need to do only a polynomial amount of work to find a much improved power gain, and present two enumeration-based pseudopolynomial algorithms that manipulators can use. Furthermore, we provide a careful investigation of heuristics for annexation which provide huge savings in computational efforts over the enumeration-based method. The benefits achievable by manipulating agents using these heuristics also compare with those of the enumeration-based method which serves as upper bound.

Acknowledgements. This work is supported by NSF research grant #0812039 entitled "Coalition Formation with Agent Leadership".

References

1. Alonso-Meijide, J.M., Bowles, C.: Generating Functions for Coalitional Power Indices: An Application to the IMF. Annals of Operations Research 137, 21–44 (2005)
2. Aziz, H., Bachrach, Y., Elkind, E., Paterson, M.: False-name manipulation in Weighted Voting Games. Journal of Artificial Intelligence Research 40, 57–93 (2011)
3. Aziz, H., Paterson, M.: False-name Manipulations in Weighted Voting Games: splitting, merging and annexation. In: 8th International Conference on Autonomous Agents and Multiagent Systems, Budapest, Hungary, pp. 409–416 (2009)
4. Aziz, H., Paterson, M., Leech, D.: Classification of Computationally Tractable Weighted Voting Games. In: Proceedings of the World Congress on Engineering, London, U.K., vol. I (2008)
5. Bachrach, Y., Markakis, E., Procaccia, A.D., Rosenschein, J.S., Saberi, A.: Approximating Power Indices - Theoretical and Empirical Analysis. Autonomous Agents and Multiagent Systems 20(2), 105–122 (2010)
6. Banzhaf, J.: Weighted Voting Doesn't Work: A Mathematical Analysis. Rutgers Law Review 19, 317–343 (1965)
7. Bilbao, J.M., Fernandez, J.R., Losada, A.J., Lopez, J.J.: Generating functions for computing power indices efficiently. TOP: An official Journal of the Spanish Society of Statistics and Operations Research 8(2), 191–213 (2000)

8. Brams, S.J.: Game Theory and Politics. Free Press, New York (1975)
9. Brams, S.J., Affuso, P.J.: Power and Size: a new paradox. Theory and Decision 7, 29–56 (1976)
10. Deegan, J., Packel, E.W.: A New Index of Power for Simple n-Person Games. International Journal of Game Theory 7, 113–123 (1978)
11. Fatima, S.S., Wooldridge, M., Jennings, N.R.: A Randomized Method for the Shapely Value for the Voting Game. In: AAMAS, Hawaii, pp. 955–962 (2007)
12. Felsenthal, D.S., Machover, M.: The Measurement of Voting Power: Theory and Practice, Problems and Paradoxes. Edward Elgar, Cheltenham (1998)
13. Felsenthal, D.S., Machover, M.: Annexation and Alliances: When Are Blocs Advantageous a Priori. Social Choice and Welfare 19(2), 295–312 (2002)
14. Ghallab, M., Nau, D., Traverso, P.: Automated Planning - Theory and Practice. Morgan Kaufmann Publishers (2004)
15. Holler, M.J., Packel, E.W.: Power, Luck and the Right Index. Journal of Economics 43, 21–29 (1983)
16. Johnston, R.J.: On the Measurement of Power: Some Reactions to Laver. Environment and Planning 10, 907–914 (1978)
17. Laruelle, A.: On the Choice of a Power Index. Instituto Valenciano de Investigaciones Economicas 2103, 99–10 (1999)
18. Lasisi, R.O., Allan, V.H.: Annexations and Merging in Weighted Voting Games - The Extent of Susceptibility of Power Indices. In: 3rd International Conference of Agents and Artificial Intelligence (ICAART), Rome, Italy, pp. 124–133 (2011)
19. Lasisi, R.O., Allan, V.H.: A Search-based Approach to Annexations and Merging in Weighted Voting Games. In: 4th International Conference of Agents and Artificial Intelligence (ICAART), Vilamoura, Algarve, Portugal, pp. 44–53 (2012)
20. Leech, D.: Voting Power in the Governance of the International Monetary Fund. Annals of Operations Research 109(1), 375–397 (2002)
21. Matsui, T., Matsui, Y.: A Survey of Algorithms for Calculating Power Indices of Weighted Majority Games. Journal of the Oper. Res. Society of Japan 43(1) (2000)
22. Matsui, Y., Matsui, T.: NP-completeness for calculating power indices of weighted majority games. Theoretical Computer Science 263(1-2), 305–310 (2001)
23. Shapley, L.S., Shubik, M.: A Method for Evaluating the Distribution of Power in a Committee System. American Political Science Review 48, 787–792 (1954)
24. Shehory, O., Kraus, S.: Methods for Task Allocation via Agent Coalition Formation. Artificial Intelligence 101(1-2), 165–200 (1998)
25. Yokoo, M., Conitzer, V., Sandholm, T., Ohta, N., Iwasaki, A.: Coalitional Games in Open Anonymous Environments. In: AAAI, pp. 509–515 (2005)

^{sp}X-Machines: Formal State-Based Modelling of Spatial Agents

Isidora Petreska[1], Petros Kefalas[2], Marian Gheorghe[3], and Ioanna Stamatopoulou[2]

[1] South East European Research Centre (SEERC),
24 Proxenou Koromila Str., Thessaloniki 54622, Greece
[2] CITY College, International Faculty of the University of Sheffield,
3 Leontos Sofou Str., Thessaloniki 54626, Greece
[3] University of Sheffield, Dept. of Computer Science,
Regent Court, 211 Portobello Str., Sheffield S1 4DP, U.K.
ispetreska@seerc.org,
{kefalas,istamatopoulou}@city.academic.gr,
m.gheorghe@dcs.shef.ac.uk

Abstract. Applications of biological or biologically inspired multi-agent systems (MAS) often assume a certain level of reliability and robustness, which is not always straightforward. Formal modelling and verification of MAS may present many interesting challenges. For instance, formal verification may be cumbersome or even impossible to be applied on models with increased complexity. On the other hand, the behaviour of MAS consists of communities evolving in space and time (such as social insects, tissues, colonies of bacteria, etc.) which are characterised with a highly dynamic structure. In order to provide a neat and effective way to modelling and verification of these systems, we focus on their spatial characteristics. Spatial agents (i.e. agents distributed and moving through a physical space) can be modelled with X-machines – one of the most prominent formalisms for modelling the behaviour of biological colonies. However, it will be demonstrated that there are certain problems in the X-machines models, common to every spatial MAS. To overcome these disadvantages, we introduce an X-machines variation that besides facilitating formal modelling, will provide grounds towards visual animation of these systems. This approach resulted into a novel progression, Spatial X-machines (^{sp}XM), without retracting the legacy characteristics of X-machines such as testing and verification strategies. Finally, we present a supporting framework to modelling and verification of spatial multi-agent system, by utilizing the Spatial X-machines approach.

Keywords: Formal modelling, X-machines, Spatial agents.

1 Introduction

Formal modelling is considered as one of the most essential stages in Multi-agent system (MAS) development and can be carried out with many different methods and techniques, ex. Z [1], VDM [2], FSM [3], Petri Nets [4], and others. These varieties of

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 379–391, 2013.

formal methods in agent-oriented engineering, might not be best for modelling biological phenomena due to the lack of expressiveness when it comes to such large-scale communicating and emergent systems. Moreover, as the complexity of a MAS increases, considerable difficulties get introduced in the process of formal modelling.

Biological and biologically-inspired systems (bio-MAS) can be characterised as spatial MAS. Spatial agents can be defined as collections of agents distributed and moving through a physical space. They have incomplete knowledge of the environment and can change their direction and position within the environment. When it comes to data modelling of spatial agents, there are three key features in their development [5]:

- Conceptual data model, which acts as a representation of the reality as it is, defined by the users;
- Logical data model, which defines how the conceptual model will be implemented; and
- The physical data model or the computer code of the application.

Targeting the conceptual data model, there are different approaches for modelling spatial phenomena of biological systems: process algebra can be applied to develop a calculus of processes that could describe the spatial geometric transformations [6]. Membrane computing can also be utilised by introducing geometric information [7] or population P systems [8]. Another agent-based approach is the intracellular NF-κB signalling pathway for modelling spatial information in predictive complex biological systems [9]. However, the combination of biological agents and spatial data modelling still remains an active research field. With this work we focus on modelling spatial agents with a formal method, namely X-machines, which are targeted at representing the behaviour of biological colonies.

Defined as complex systems, spatial MAS should be characterised with reliability, quality and robustness. Therefore, besides modelling, verification can be considered the next important step in the developing cycle. However, it is fairly straightforward (although not easy) to apply the known verification and validation techniques to spatial systems composed from one agent, but it is extremely hard to transfer such techniques to spatial MAS with a dynamic behaviour. We discuss that visual animation as an informal verification technique, helps in discovering the flaws of the formally unverifiable properties within a spatial MAS (such as their position or direction with respect to the environment). It is argued that attempts to formally verify such properties would result into a combinatorial explosion. Moreover, visual animation provides means to facilitate the communication gap between the formal methods experts and the biologists (which in turn have no formal background) by providing an immediate feedback understandable to both sides. And finally, visual animation can often lead to detecting some emergent behaviour within a system. Therefore, this work opens the question about how to automatically visualise a given state based model.

Starting with a detailed discussion on X-machines, later extended by a case study (Section 2), we demonstrate its drawbacks when it comes to modelling the spatial characteristic of an agent. This provides grounds to introducing a formal variation called Spatial X-machines (^{sp}XMs, Section 3). This structure will be carefully deliberated, followed by discussion about their support towards verification and validation (Section 4). Given a formal model, it is interesting to consider what properties should be model checked, such as whether there is an emergent behaviour in the system when this MAS is massively populated with similar agents. This idea is captured in Section 5, which leads to the development of a supporting framework for modelling and verification of bio-MAS. Finally, a discussion and ideas for future work conclude the paper.

2 Background on X-Machines: Theory, Definition and Practice

An X-machine (XM) resembles a Final State Machine (FSM) with the power of being more expressive [10]. This is achieved due to the addition of a memory, and functions operating on the inputs and memory values. Considering stream XMs (a representative class of XMs), the memory is a typed tuple of values which supports the modelling of complex data structures. The control, on the other hand, can be visualised by utilising a diagrammatic approach. Thus stream XMs have the ability of modelling both the data (held in the memory) and the control. The processing of the data is modelled by transitions between states, represented with functions. A function receives the memory values together with an input, performs changes on these memory values and produces an output. Based on the current state and application of a function, the stream XM evaluates the next state.

Formally, a stream XM can be described as an 8-tuple XM = (Σ, Γ, Q, M, Φ, F, q_0, m_0), such that [11, 12]:

- Σ and Γ are the sets of input and output symbols, respectively;
- Q is a finite set of states;
- M is an n-tuple called memory;
- Φ is a finite set of partial functions that map an input and a memory state to an output and a new memory state, ϕ: $\Sigma \times$ M $\rightarrow \Gamma \times$ M;
- F is a function that determines the next state, given a state and a function from the type Φ, F: Q $\times$ Φ $\rightarrow$ Q, for deterministic XMs; and
- q_0 and m_0 are the initial state and memory respectively.

With the focus on the practical development of communicating systems, the output of an X-machine function can become input to a function of another X-machine. This way a structure known as Communicating X-machine (CXM) is being formed, providing a way to deal with agents communication [13, 14].

2.1 Case Study: A Foraging Agent

Let us consider the following example of an agent that randomly moves in 2-D space, picks up an object it encounters and carries it back to the base (Fig. 1). Clearly, although this is a very simple example, there could be quite a few solutions (from a very abstract to more detailed one). Likewise, Fig. 2 depicts three alternative XM models that can be considered as solutions to the foraging agent problem.

Table 1 demonstrates three ways of modelling the foraging agent problem (the numbering (a), (b) and (c) corresponds to the numbering in Fig. 2). Solution (a) is a very abstract representation that does not even take into consideration the position coordinates of the agent. The fact that XMs are generic and do not impose modelling of

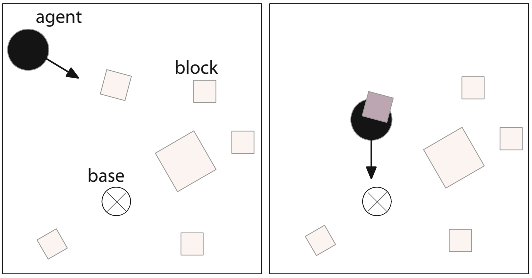


Fig. 1. The foraging agent example

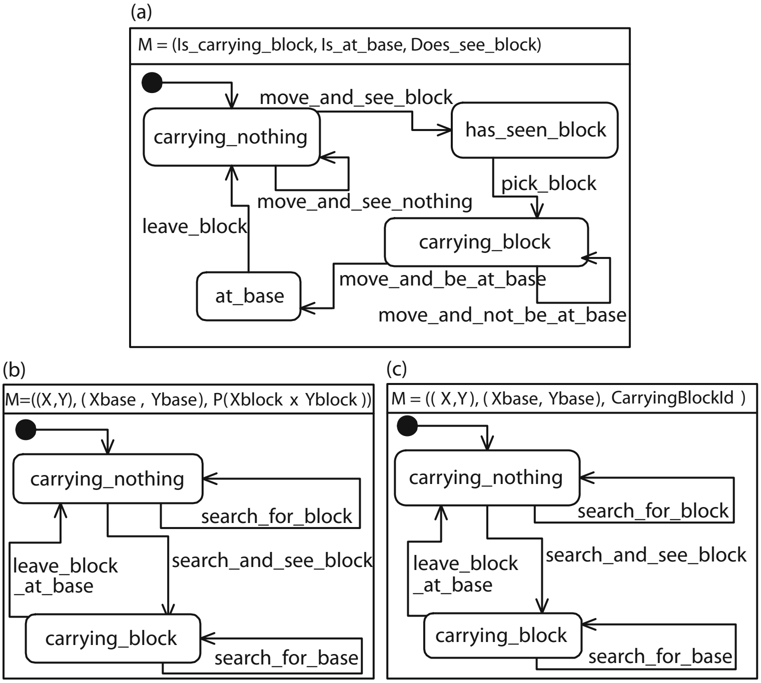


Fig. 2. Examples of modelling the foraging agent example: a) very abstract representation b) more detailed, but complex representation c) the best represented solution

a position, in such an example might result into an incomplete model. A more detailed representation can be derived from solution (b) (it can be noted that the representation is a design choice, for instance the memory variables that correspond to positions are integers). Yet again, this representation is way too complex and probably more difficult to work with. Finally, the last representation (c), is the best solution in terms of completeness and complexity because it deals with the positioning of the agent (adds to completeness), but does not deal with the positioning of the blocks (reduced complexity).

Table 1. Different ways of modelling the foraging agent example

(a)	Q = {carrying_nothing, has_seen_block, carrying_block, at_base} M = (Is_carrying_block, Is_at_base, Does_see_block), where Is_carrying_block, Is_at_base, Does_see_block ∈ {true, false} m_o = (false, false, false) q_o = carrying_nothing Σ = {“move to a place w/o block”, “move to a place with block”, “pick block”, “search for base”, “move to base”, “leave block”} Γ = {“agent keeps moving empty”, “agent detected block”, “agent picked block”, “agent searches for base”, “agent found base”, “agent left block”}
(b)	Q = {carrying_nothing, carrying_block} M = ((X, Y), (X_{base}, Y_{base}), $\mathbb{P}(X_{block} \times Y_{block})$, Hand) where X, Y, X_{block}, Y_{block}, X_{base}, Y_{base} ∈ $\mathbb{Z}$, Hand ∈ {full, empty} m_o = ((2, 3) , (0, 0), {(2, -3), (4, -6), (2, 1), (3, 5), (-1, 5)}, empty) q_o = carrying_nothing Σ = (X, Y), where $X, Y \in \mathbb{Z}$ Γ = {“agent keeps moving empty”, “agent detected and picked block”, “agent searches for base”, “agent found base and left block”}
(c)	Q = {carrying_nothing, carrying_block} M = ((X, Y), (X_{base}, Y_{base}), CarryingBlockId) where X, Y, X_{base}, Y_{base} ∈ $\mathbb{Z}$, CarryingBlockId ∈ {$block_1$, $block_2$, ... $block_n$} ∪ nil, n ∈ $\mathbb{N}$ m_o = ((2, 3), (0, 0), nil) q_o = carrying_nothing Σ = ((X, Y), BlockId), where X, Y ∈ $\mathbb{Z}$, BlockId ∈ {$block_1$, $block_2$, ... $block_n$} ∪ {nil}, n ∈ $\mathbb{N}$ Γ = {“agent keeps moving empty”, “agent detected and picked block”, “agent searches for base”, “agent found base and left block”}

2.2 Shortcomings of XM for Spatial Agents

The case study in the previous section demonstrated that there might be different ways to modelling spatial agents with the XM approach, even for the simplest scenario. This leads towards the identification of the following shortcomings when modelling spatial agents with XM:

- There might be many different solutions, even for the simplest model, for representing the commonly found properties, such as the initial position or the direction of a spatial agent. This makes it more difficult to understand a given model (we have to understand how the modeller decided to represent these properties), and to create one (the modeller has to decide how to represent these properties on any given case).
- There are difficulties in simulating a given model because there is not a standard way that deals with the manipulation and processing of the spatial properties like the initial position or the direction of a spatial agent.
- The memory holds all data structures required, including the position and the direction.

Initiated by these shortcomings, a question that can be imposed is: *How can we redefine XMs to support spatial agent modelling natively?* The motivation behind this question can be further broadened into the following aspects:

- The subset of MAS that deals with movement in space is quite numerous, starting with bio-MAS, up to MAS used in many industrial applications like robotics, etc.
- Different modellers might represent a spatial agent's basic characteristics, like position and direction, in different ways.
- The current XM representation for a spatial agent model does not directly map to an animation/simulation.
- The current XM representation for a spatial agent model is rather cumbersome/ difficult to code, and in many situations it is also difficult to be understood.
- When it comes to verifying a spatial model with XM, this will result into space explosion due to the spatial information.

3 Introduction to Spatial X-Machines

^{sp}XMs represent an extension of stream XMs, which define three new components and modify some existing ones in order to facilitate unification with the newly defined components. The input and the output set, the memory, the set of states and the next state remain intact, because these structures do not deal with the spatial attributes that we intend to support. On the other hand, the following three components have been introduced:

- A tuple containing the current position of the agent, which determines the agent's location in its environment.
- An integer that represents the agent's current direction (such as south, north, etc.) that represents its heading.
- A set containing elementary operations. These operations allow manipulation of the current position and the current direction.

An ^{sp}XM is a 13-tuple ^{sp}XM = (Σ, Γ, Q, q_0, π, π_0, θ, θ_0, M, m_0, E, Φ, F) that can be formally defined as:

- Σ is the set of input symbols;
- Γ is the set of output symbols;
- Q is a finite set of states;
- q_0 is the initial state;
- M is an n-tuple called memory;
- m_0 is the initial memory;
- π is a tuple of the current position, i.e. (x, y) when a 2D representation is considered;
- π_0 is the initial position;
- θ is an integer in the range 0 to 360, that represents a direction (integer values are used as a design choice);
- θ_0 is the initial direction;
- E is a set which contains elementary positioning operations: $e_i : \Pi \times \Theta \longrightarrow \Pi \times \Theta$, such as direction change, moving forward and moving to a specific position;

- Φ is a finite set of partial functions ϕ that map a memory state, position, direction and set of inputs to a new memory state, position, direction and set of outputs: ϕ: M $\times$ π $\times$ θ $\times$ Σ $\longrightarrow$ M $\times$ π $\times$ θ $\times$ Γ; and
- F: Q $\times$ Φ $\rightarrow$ Q is a function that determines the next state, given a state and a function from the type Φ.

If we take a closer look at the memory M, it may be noted that M is composed of $M'^{\wedge}< \pi,\ \theta >$, where M' is a memory structure from the standard XM. Moreover, regarding the set of elementary operations, we have currently defined three operations:

- change_direction m - changes the spatial agent's direction to m, where m is of type θ and $m \in \Theta$ (ex. *change_direction 60*)
- move_n_forward n - moves forward for n units, where n is an integer (ex. *move_n_forward 3*)
- move_to_position *x y* - moves to specific position (*x*, *y*), where *x* is the x-coordinate, *y* is the y-coordinate of the agent and (*x*, *y*) $\in \Pi$ (ex. *move_to _position 126 43*)

As it can be determined from the definition, an ^{sp}XM in essence provides a separation of the behaviour within the system that deals with the movement (and the other spatial attributes) from the rest of the behaviour. This way we establish a standardised way to modelling motion that is easily understandable and provides a direct mapping to an animation/simulation. And finally, this definition maintains an obvious equivalence with the standard XM.

4 Verification and Validation Strategies

Formal verification of spatial agents is an extremely complex task. On one hand stands the fact that the verification process leads to combinatorial explosion, because modelling these agents means modelling of their spatial properties. Therefore, the verification would require exploration of a state space developed by the combination of all agent positions evolved through time [15]. On the other hand, there is the fact that the emergent properties of the system should be known in advance in order to be verified in silico. The concept of emergence can be explained as a pattern appearing in the configuration of the agents, at some instance during the lifetime of the system. In bio-MAS the emergence can be observed in-vivo (for example, line formation, flocks, schools, herds etc.). However, when it comes to artificial agents, it is not always straightforward. Driven from these two problems, it might be desirable to combine several formal with informal techniques that would be able to join forces towards the verification of spatial MAS [15].

4.1 Verification and Validation of ^{sp}XMs

The models of an X-machine can be described with the X-machine Definition Language (XMDL [11]) and textual simulation. XMDL is a listing of definitions that match the tuples of X-machine's definition. Similar to the diagram of Fig. 3 (which is essence targets ^{sp}XMs), XMDL is facilitated with a parser built using Definite Clause Grammars

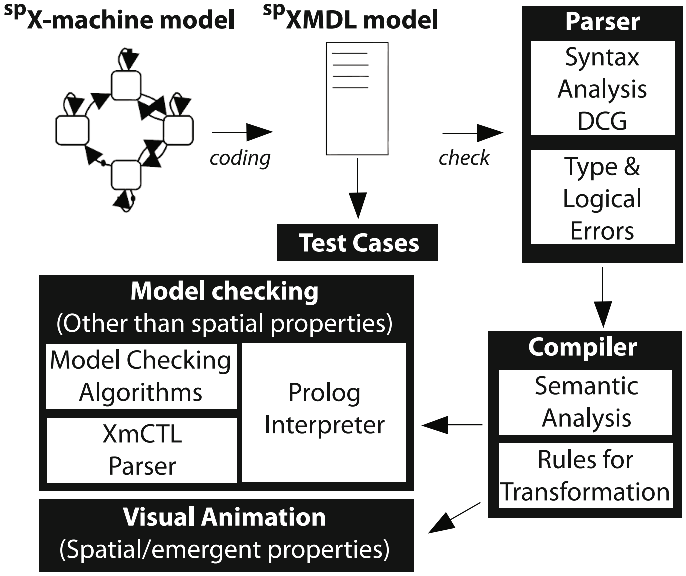


Fig. 3. Verification and validation of an ^{sp}X-machine

(DCG) notation [10], which apart from the syntax errors, output as warnings any kind of logical errors or omissions. The semantic analysis and the rules for transformation are being checked by the compiling component, with the aid of defined rules under which the specification is translated into the equivalent Prolog code. This Prolog code is after utilised by an animation tool, which simulates the computation of an X-machine. Furthermore, the model checking component defines a new logic, XmCTL [16], and with the implementation of a model checking algorithms can determine whether a property is true or false. And finally, XMs are supported with automatic generation of test cases, which is proved that finds all faults in the implementation [17].

Taking ^{sp}XMs into consideration, the following discussion will concentrate on investigating whether they inherit the mentioned verification and validation techniques of XMs. An informal proof that an ^{sp}XM is equivalent to any XM could be derived by investigating:

- The memory M of a normal XM is equivalent to the structure of memory M, position Π and direction Θ within an ^{sp}XM. In other words, the position and direction can either become members of the memory tuple in a normal XM model, or they can be excluded from the model without loss of its integrity.
- Any function in an ^{sp}XM model can be translated into a function of the normal XM. More particularly, the predefined spatial operations of a function in an ^{sp}XM model can be omitted or replaced with the standard XMDL syntax to preserve the logic flow.

Along these lines, by removing the newly defined components that in essence define an ^{sp}XM, what we get is still a completely valid skeleton of a normal XM model. Therefore, ^{sp}XMs tend to provide a standardised way of representing some properties of the system, which could also be represented with an XM model and this can lead in easy formalisation, verification (model checking), testing and implementation. The only condition imposed would be not to test or model check the position (coordinates) and direction properties, which in turn will result into state explosion.

4.2 Simulation of ^{sp}XMs

Referring to the initial discussion of this chapter for combining formal with informal techniques towards the verification of spatial MAS, we suggest that visual animation can be exploited for detecting the emergent properties of a system. In bio-MAS, animation becomes even more interesting because of the spatial attributes of an agent, e.g. agents move in an n-dimensional space. An animator as a form of simulation, is any kind of program which given the code in the intermediate language, implements an algorithm to facilitate the computation of the model and its output though a textual description [18, 19]. However, most of the animation techniques share one major drawback, i.e. the outputs they produce are in a textual form and thus not even close to the real visual perceptions on the system. Therefore, we focus on a visual simulation platform, namely NetLogo [18, 20].

NetLogo is a simulation platform for visual animation of multi-agent systems supported by a functional language that can represent an agent's behaviour, as well as by an environment for the creation of a graphical user interface. NetLogo facilitates the verification of a biological model in a way that simulation scenarios may be executed, and thus the expected behaviour of the system could be compared to the visual outcome. This platform was our initial choice due to its simplicity and the legacy of work we have done so far in experimenting with Netlogo features and emergent biological phenomena. Similar but more advanced development toolkits such as Repast [21] should also be considered as alternatives to visualisation [22].

Given an XM model, it is not always easy nor uniform to map its representation into the equivalent NetLogo code. This is due to the already discussed disadvantages in Sec. 3 that deal with the behaviour of the system that represents the motion (and the other spatial attributes). This raises the question: *Having a model of a system, how can we visualise it?* ^{sp}XMs overcome the problem found in XMs models, and thus enhance visual animation, as the agent's position and direction can be interpreted into motion within an animation platform. This feature opened the horizon towards ideas for automation of the simulation scenarios for an ^{sp}XM model, resulting into a tool ^{sp}XM2Visual.

As it can be noticed from Fig. 4, the ^{sp}XM2Visual system architecture consists of two main components, the parser (notifies of possible errors, like type and logical ones) and the compiler (contains all the rules and the logic for the translation).

Given that NetLogo supports only lists (this is a mathematical structure similar to the array found in a programming language), in order to produce an equivalent NetLogo model from an ^{sp}XM representation, there was a need of creating an external library. This external library for NetLogo (included in the compiler of Fig. 4) supports all the mathematical primitives found in ^{sp}XMDL (sets, bags, sequences, etc.) and their operations (see [23] for examples). Moreover, the set of operations from ^{sp}XMs is translated within this library as functions (an agent's movement to a certain position, the

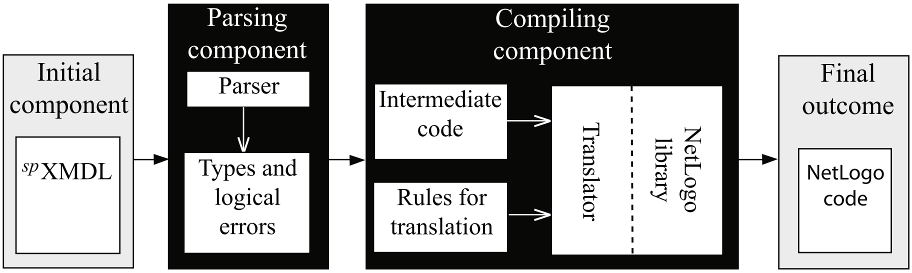


Fig. 4. System architecture of ^{sp}XM2Visual

perception of the environment, etc.). Functions that deal with the modelling of the environment (defining obstacles, defining agents, etc.) are also included. Other interesting parts of the compiling component are the *rules for transformation* and the *translator*. The rules for transformation are simple *if-then rules*. An example of such a rule for the *where statement* in an ^{sp}XMDL representation is presented in [23]. On the other hand, the translator is composed of a reader (parses an ^{sp}XMDL representation and constructs the necessary objects), an object model (function, memory, state, transition, etc.) and a writer (used in writing the NetLogo code).

5 A Proposed Framework towards Verification of Spatial MAS

The concept of emergence can be explained as a pattern appearing in the configuration of the agents, at some instance during the lifetime of the system. We are interested in the type of emergence related to the positioning in space (such as line formation, flocks, schools, herds etc.). However, formal verification techniques can only be applied under the assumption that we know what is the emergent property. In order to be able to tackle this kind of problems, we propose a research framework, depicted in Fig. 5. This framework helps in identifying emergent behaviour through the automatic transformation of a formal model to an executable visual simulation [15].

The frame on the right side of the figure, reflects the current work. At the top of the framework is the phase of formal modelling of agents. For this step we utilise the ^{sp}XMs approach, because the formal models should be able to clearly distinguish modelling of various types of behaviours (such as spatial or other behaviours, communication, dynamic organisation etc.). This makes it possible to apply different transformations that will facilitate further processing. On one hand, the spatial behaviour can lead towards visual animation which can help in detecting emergence. On the other hand, abstractions of the spatial behaviour together with the rest of the behaviours can lead towards simulation and logging of time series data [15]. Finally, all the patterns of behaviours together with the visual animation would produce a set of desired properties. The properties can be verified in the original spatial agent model by model checking.

The spatial behaviour can be detected by utilizing NetLogo with the automatic translation tool of an ^{sp}XM model. On the other hand, the logging of time series data might be accomplished with a tool such as FLAME [24, 25], used to animate XM models. The

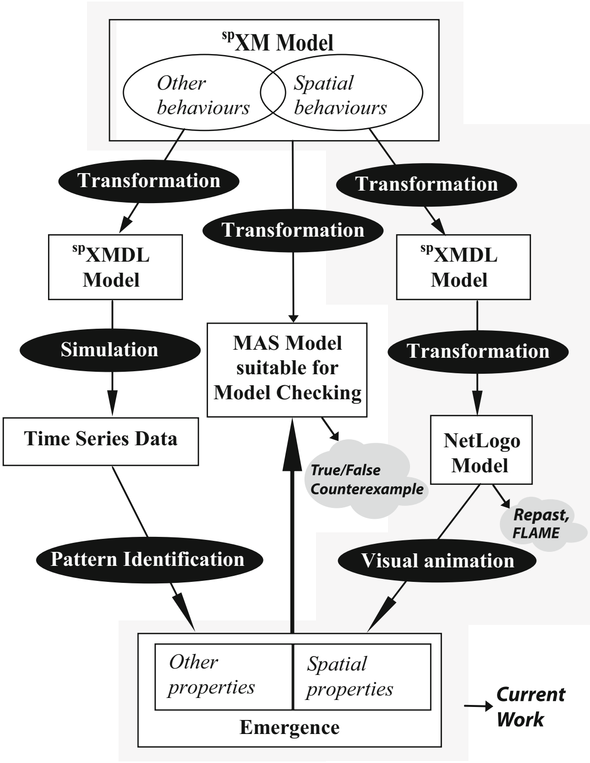


Fig. 5. A framework for validating emergent properties in spatial biology-inspired MAS

next step involves utilizing a tool for identifying patterns, such as DAIKON [26]. Finally, the ^{sp}XM can be suitably transformed into an equivalent model in SPIN, PRISM or SMV [27–29]. In this case, given a temporal formulae, all of the desired properties could be verified upon the original model. More information about this framework can be found in [15].

6 Conclusions and Future Work

Given X-machines, one might argue that the bio-MAS models might be very abstract, i.e. there is a freedom in the representation of a model. On the other hand, certain knowledge is required for simulating a biological agent, for instance the initial position or direction of the agent. This introduces difficulties in simulating a given model, because an X-machine does not specify how these knowledge will be modelled. Thus we presented an idea of extending X-machines into more specific formalism for modelling spatial agents that move in space, called ^{sp}XMs. Unlike other formalisms that go behind the concept of treating the agent's behaviour as one uniform component, Spatial X-machines tend to draw a separation between different types of agent's behaviour. This approach facilitates the verification of emergent behaviour of spatial MAS fitting perfectly in the proposed framework towards verification of spatial MAS.

Further work includes extending ^{sp}XMs in a way to support other spatial agent properties and more functions that will bring in more realistic modelling of the spatial concept. Examples could also be created for simulation, testing and model checking of XM and ^{sp}XM models. This can lead to a critical comparison of their differences and advantages. Regarding the tool for automatic translation of an ^{sp}XM model into the NetLogo platform, some improvements in the compilation process can be considered. Finally, applying real-world case studies could greatly boost confidence on the proposed framework.

References

1. Spivey, M.: The Z Notation: A Reference Manual. Prentice-Hall, Englewood Cliffs (1989)
2. Jones, C.B.: Systematic Software Development using VDM, 2nd edn. Prentice-Hall, Englewood Cliffs (1990)
3. Holcombe, M.: X-machines as a basis for dynamic system specification. Software Engineering Journal, 69–76 (1988)
4. Reisig, W.: Petri nets: An introduction. EATCS Monographs on Theoretical Computer Science. Springer, Berlin (1985)
5. Pageau, J., Bédard, Y., Caron, C.: Spatial data modeling: The Modul-R formalism and CASE technology. In: ISPRS Symposium, Washington, United-States, August 1-14 (1992)
6. Cardelli, L., Gardner, P.: Processes in Space. In: Ferreira, F., Löwe, B., Mayordomo, E., Mendes Gomes, L. (eds.) CiE 2010. LNCS, vol. 6158, pp. 78–87. Springer, Heidelberg (2010)
7. Romero-Campero, F.J., Twycross, J., Camara, M., Bennett, M., Gheorghe, M., Krasnogor, N.: Modular assembly of cell systems biology models using P Systems. International Journal of Foundations of Computer Science, 427–442 (2009)
8. Petreska, I., Kefalas, P.: Population p systems with moving active cells. In: Gheorghe, M., Păun, G., Verlan, S. (eds.) Twelfth International Conference on Membrane Computing (CMC12), pp. 421–432. Laboratoire d'Algorithmique Complexité et Logique of the University of Paris Est – Créteil Val de Marne, Fontainebleau (2011)
9. Pogson, M., Holcombe, M., Smallwood, R., Qwarnstrom, E.: Introducing spatial information into predictive NF-kB modelling – An agent-based approach. PLoS ONE 3(6), e2367 (2008)
10. Kefalas, P., Eleftherakis, G., Sotiriadou, A.: Developing tools for formal methods. In: Proceedings of the 9th Panehellenic Conference in Informatics (2002)
11. Kefalas, P., Kapeti, E.: A design language and tool for X-machines specification. In: Fotiadis, D.I., Nikolopoulos, S.D. (eds.) Advances in Informatics, pp. 134–145. World Scientific Publishing Company, Singapore (2000)
12. Kefalas, P., Holcombe, M., Eleftherakis, G., Gheorge, M.: A formal method for the development of agent based systems. In: Plekhanova, V. (ed.) Intelligent Agent Software Engineering, pp. 68–98. Idea Group Publishing Co. (2003)
13. Kefalas, P., Eleftherakis, G., Kehris, E.: Communicating X-machines: A practical approach for formal and modular specification of large systems. Information and Software Technology 45, 269–280 (2003)
14. Kefalas, P.: Formal Modelling of Reactive Agents as an Aggregation of Simple Behaviours. In: Vlahavas, I.P., Spyropoulos, C.D. (eds.) SETN 2002. LNCS (LNAI), vol. 2308, pp. 461–472. Springer, Heidelberg (2002)

15. Petreska, I., Kefalas, P., Gheorghe, M.: A framework towards the verification of emergent properties in spatial multi-agent systems. In: Ivanovi, M., Ganzha, M., Paprzycki, M., Badica, C. (eds.) Proceedings of the Workshop on Applications of Software Agents, pp. 37–44. Department of Mathematics and Informatics Faculty of Sciences, University of Novi Sad, Serbia (2011)
16. Eleftherakis, G., Kefalas, P.,, S.: XmCTL: Extending temporal logic to facilitate formal verification of X-machines. Matematica-Informatica, 79–95 (2002)
17. Ipate, F., Holcombe, M.: Specification and testing using generalised machines: a presentation and a case study. In: Software Testing, Verification and Reliability, pp. 61–81 (1998)
18. Wilensky, U.: NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern Univ., Evanston, IL (1999), `http://ccl.northwestern.edu/netlogo/`
19. Stamatopoulou, I., Kefalas, P., Gheorghe, M.: OPERAS: A Framework for the Formal Modelling of Multi-Agent Systems and Its Application to Swarm-Based Systems. In: Artikis, A., O'Hare, G.M.P., Stathis, K., Vouros, G.A. (eds.) ESAW 2007. LNCS (LNAI), vol. 4995, pp. 158–174. Springer, Heidelberg (2008)
20. Wilensky, U.: NetLogo Segregation model. Center for Connected Learning and Computer-Based Modeling, Northwestern Univ., Evanston, IL (1997), `http://ccl.northwestern.edu/netlogo/models/Segregation`
21. Collier, N.T., North, M.J.: Repast SC++: A platform for large-scale agent-based modeling. In: Large-Scale Computing Techniques for Complex System Simulations. Wiley (2011) (in press)
22. Petreska, I., Kefalas, P., Gheorghe, M.: Tools for simulating spatial mas. In: Proceedings of the 7th Annual SEERC Doctoral Student Conference, DSC 2012 (2012) (in print)
23. Petreska, I., Kefalas, P., Gheorghe, M.: Informal verification by visualisation of state-based formal models. In: Proceedings of the 6th Annual SEERC Doctoral Student Conference, DSC 2011, Thessaloniki, Greece, pp. 309–319 (September 2011)
24. Qwarnstrom, E., Pogson, M., Smallwood, R., Holcombe, M.: Formal agent-based modelling of intracellular chemical interactions. Biosystems 85, 37–45 (2006)
25. Holcombe, M., Smallwood, R., Walker, D.: Development and validation of computational models of cellular interaction. Journal of Molecular Histology 35, 659–665 (2004)
26. Michael, D.E., William, G.G., Yoshio, K., Notkin, D.: Dynamically discovering pointer-based program invariants. Technical Report UW-CSE-99-11-02, University of Washington Department of Computer Science and Engineering, Seattle, WA (November 1999) (revised March 17, 2000)
27. Holzmann, G.J.: The model checker spin. IEEE IFans. on Software Engineering, 279–295 (1997)
28. Kwiatkowska, M., Norman, G., Parker, D.: PRISM: Probabilistic Symbolic Model Checker. In: Field, T., Harrison, P.G., Bradley, J., Harder, U. (eds.) TOOLS 2002. LNCS, vol. 2324, pp. 200–204. Springer, Heidelberg (2002)
29. McMillan, K.L.: Symbolic Model Checking. Kluwer Academic Publishers, Englewood Cliffs (1993)

Talking Topically to Artificial Dialog Partners: Emulating Humanlike Topic Awareness in a Virtual Agent

Alexa Breuing and Ipke Wachsmuth

Artificial Intelligence Group, Bielefeld University, Bielefeld, Germany
{abreuing,ipke}@techfak.uni-bielefeld.de

Abstract. During dialog, humans are able to track ongoing topics, to detect topical shifts, to refer to topics via labels, and to decide on the appropriateness of potential dialog topics. As a result, they interactionally produce coherent sequences of spoken utterances assigning a thematic structure to the whole conversation. Accordingly, an artificial agent that is intended to engage in natural and sophisticated human-agent dialogs should be endowed with similar conversational abilities. This paper presents how to enable topically coherent conversations between humans and interactive systems by emulating humanlike topic awareness in the virtual agent Max. Therefore, we firstly realized automatic topic detection and tracking on the basis of contextual knowledge provided by Wikipedia and secondly adapted the agent's conversational behavior by means of the gained topic information. As a result, we contribute to improve human-agent dialogs by enabling topical talk between human and artificial interlocutors. This paper is a revised and extended version of [1].

Keywords: Automatic topic awareness, Embodied conversational agents, Human-agent interaction, Topic detection and tracking, Wikipedia.

1 Motivation

Topic awareness plays an important role in human conversations. Besides resolving linguistic references and ambiguities which often arise in natural language talks, it enables the interlocutors to interactionally produce coherent sequences of spoken utterances. More precisely, every spoken contribution may raise new potential topics whose actual realization depends on the co-participant's acceptance by picking up one of these topics within his or her reply [2]. Hence, a topic can be described as a *joint project* [3] as it is jointly established during ongoing conversations. Furthermore, being aware of topics helps us to touch the right subject according to the social circumstances enclosing the interactional situation. Assuming an everyday small talk conversation, for example, so-called unsafe topics such as religion and death, should be avoided [4]. Altogether, the competence to talk topically constitutes a basic requirement to carry on meaningful, flexible, and appropriate conversations with other persons.

Embodied conversational agents (ECAs) are virtual characters possessing humanlike conversational behaviors to establish an intuitive human-machine interface [5]. That is, they are capable of holding face-to-face conversations with humans by understanding

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 392–406, 2013.

and producing speech, gestures, and facial expressions. Nevertheless, they often fail to converse in great depth and hence to mutually establish a topical talk with their human opponent. In addition, many ECAs lack in simulating a sense for the adequacy of certain topics during dialog. To remedy these weaknesses, the artificial interlocutor needs to be aware of ongoing and potential conversational topics like humans.

To provide conversational agents with artificial, humanlike topic awareness in everyday interactions two main tasks need to be automatized: First, the detection of topics raised in ongoing natural language dialogs and second, the adequate integration of the resulting topic information into the agent's underlying system architecture. This paper introduces an approach tackling both tasks: We show how to connect well-established linguistic information retrieval methods with benefits originated from collaborative work provided by Wikipedia to automatically detect dialog topics. Additionally, we present how to utilize the obtained information to improve the conversational abilities of virtual computer characters regarding topic handling.

The rest of the paper is organized as follows. In the next section we introduce our notion of dialog topics establishing the basis for the present work. Subsequently, the several processes of our automatic topic detection approach are described in Section 3. Thereby we especially emphasize the application of collaborative knowledge provided by Wikipedia. Section 4 highlights the embedding of the resulting topic information into the existing architecture of the conversational agent Max. As a result, we contribute in emulating humanlike topic awareness in artificial agents as described by means of our dialog scenario in Section 5. Moreover, we present how to evaluate our model in the near future. In Section 6 we give an overview of related work before closing the paper with a short conclusion and discussion.

2 Introducing Dialog Topics

Assuming dialogs to be face-to-face conversations between two partners, a dialog topic emerges from a *joint activity* performed by both interlocutors [6]. That is, considering single utterances to specify a dialog topic is insufficient as they do not have topics in isolation. They rather provide topic suggestions [7]. However, the topic formulation of the particular topic is done at different levels of abstraction and from different subjective positions [2]. Speaker A, for example, might categorize a dialog about Whiskey and Brandy using the term "Alcohol", whereas speaker B might choose the term "Drinks" or "Spirits" referencing the same topic. According to this, we define a dialog topic to be an *independent, self-selected category superordinate to a co-constructed sequence of dialog contributions* [8].

2.1 Topic Shifts

A dialog topic subordinates a sequence of coherent dialog contributions as wholes [9,2]. Hence, they generalize the concepts mentioned in these contributions to a certain degree. A potential *topic shift* in dialogs occurs, once previous concepts and concepts coming up subsequently cannot be generalized to one topic anymore. If attending to the new concepts opens a completely different dialog topic and comes along with a drop of the present one, we refer to this kind of shift as *topic leap* [2].

On the other hand, a topic shift might happen gradually. Imagine the following dialog sequence:

A: *"In which city do you live?"*
B: *"Munich."*
A: *"Ah, then you are a fan of Bayern Munich?"*
B: *"Actually no. I like Arsenal."*

By mentioning the concept "city", speaker A suggests to talk about places. Speaker B agrees to this topic by replying with an utterance containing the concept "Munich" specifying a German city. "Munich" in turn is unrelated to the upcoming topic "Sports", however, it is conceptually closely connected to Munich's local soccer club "Bayern Munich". Thus, the dialog merges seamlessly from the topic "Places" to the topic "Sports". Hobbs calls this phenomenon *topic drift* [10].

2.2 Selection of Dialog Topics

Raising an issue requires choosing a dialog topic first. Thereby, the amount of possible topics is constrained due to the given dialog scenario, the personal relation between the dialog partners, and their cultural background. Accordingly, not every dialog topic is appropriate for everyday small talk conversations.

Referring to Schneider [11], there are three groups of basic options for topic selection:

(1) The **immediate situation** involves all topics addressing the *frame* of the dialog situation.
(2) The **external situation** represents the larger context of the immediate situation and hence of its topics.
(3) The **communication situation** refers to the conversation partners and holds private topics such as hobbies or family.

A typical small talk starts with a topic related to the immediate situation and continues with topics from the external or communication situation. Due to these social conventions, most small talk structures are very similar and ease striking up a conversation with other, especially unknown persons.

3 Automatic Detection of Dialog Topics

Constituting a matter of course for humans, the automatic detection of dialog topics poses a great challenge. Given a dialog situation as defined before, it has to meet several requirements. First of all, the underlying processes have to work *online*. As dialogs are continuous and demand adaptive moment-by-moment decisions, it is necessary to incessantly provide the system with information about the current topic situation directly influencing the agent's conversational behavior. Additionally, this information has to be processed within a short time frame to guarantee humanlike reaction time. Moreover, the wide range of possible topics, for example being discussed in everyday conversations, calls for a *dynamic* handling of previously unknown contributions. This in turn presumes an access to huge amounts of previously unlearned topics and how they are correlated. According to the dynamic factor and for further reasons assigned subsequently, the online encyclopedia Wikipedia proved to be the ideal knowledge source.

3.1 Topics Provided by Wikipedia

According to our definition, dialog topics are considered to be *categories* subordinating a sequence of dialog contributions. The Wikipedia category system is composed of categories subordinating articles presented by natural language texts. Utilizing the similarity between utterance-topic relations in dialogs and article-category links in Wikipedia constitutes the basis for our dynamic topic detection approach. Generally speaking, we identify a dialog topic by mapping the several utterances to Wikipedia articles and specifying their shared Wikipedia categories as potential topics. Thus, the detection process is capable of identifying a topic t without having a priori knowledge of the domain underlying t.

A big advantage of accessing Wikipedia for our purpose is the fact that its encyclopedic knowledge is constructed collaboratively by numerous volunteers. Hence, Wikipedia provides huge amounts of information whose maintenance is done by others. Furthermore, the resulting description and categorization of concepts reflect the participants' perception of conceptual structures and delivers insights into the human understanding of topics and their relations.

3.2 Online Detection

Within our approach, realizing an automatic topic detection mainly involves the implementation of automatic processes that identify potential topics, track ongoing topics, detect topical shifts, and label the coherent dialog sequences. To ensure an online working topic detection the first two tasks need to be performed continuously, that is on every incoming utterance. Their outcomes simultaneously affect the remaining processes. In the following, the several tasks are described in more detail. Additionally, Figure 1 gives an overview of the presented topic detection approach and illustrates the relations between its associated processes.

Identification of Potential Topics. Referring to Schank (1977), an utterance said in response to an input provides both a conceptual intersection to the present dialog topic and a new conceptualization introducing potential new topics. Accordingly, to automatically identify potential topic directions, at first every single dialog contribution has to be conceptualized by identifying its contained *concept terms*. Therefore, the system first preprocesses the present utterance by means of the *Stanford Part-Of-Speech Tagger* [12]. Afterwards, all identified nouns and proper nouns are specified as concept terms. Moreover, the system extracts the verbs contained in the present utterance and transforms them to their substantive as providing potential conceptual information as well. Therefore we make use of the online dictionary *Wiktionary*. Then, the system searches for a Wikipedia article giving a concept description for the substantive. If a corresponding article can be found, as for example given for the term *"swimming"*, the substantive

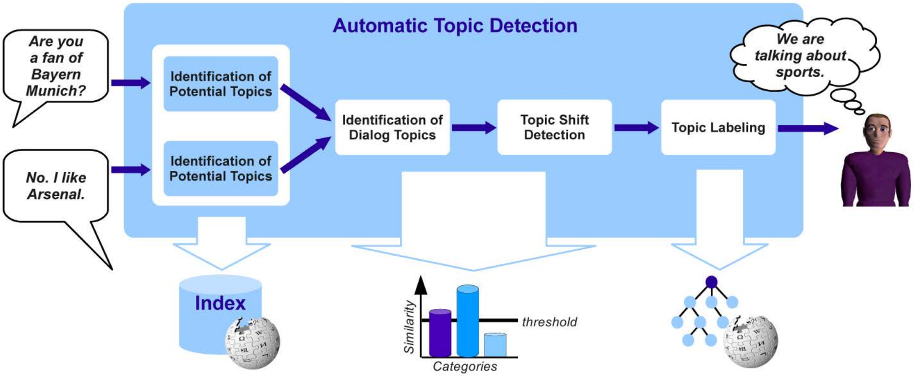

Automatic Topic Detection
Are you a fan of Bayern Munich?
No. I like Arsenal.
Identification of Potential Topics
Identification of Potential Topics
Identification of Dialog Topics
Topic Shift Detection
Topic Labeling
We are talking about sports.
Index
Similarity
threshold
Categories

Fig. 1. Overview of the processes involved in our automatic topic detection approach

is considered as a concept term furthermore. In case no article is found, the substantive is not considered as a concept term as probably not providing conceptual information (like the term *"doing"*). In addition, auxiliary verbs such as *"having"* are excluded in the first place by filtering all concept terms based on a predefined stoppword list.

In order to detect named entities consisting of more than one word, adjectives and/or nouns, and proper nouns appearing successively are tested for their lexical "togetherness". Therefore we make use of the concept information provided by Wikipedia in terms of single articles [13]. More precisely, each of these potential named entities are mapped onto the set of all Wikipedia articles A_{wiki} twice: once as a whole and once noun-wise. This mapping process is accomplished via a mapping function

$$f : cterm \mapsto A_{wiki} \quad (1)$$

where $cterm$ is either the potential named entity or a single noun. To realize f, we built up an *Apache Lucene* [14] search index containing documents for every Wikipedia article including information about their titles, textual descriptions, textual anchors of their incoming links, and redirects. This allows us to estimate both mappings by means of the Lucene similarity score

$$score(q,d) = \Sigma_{t \in q}(tf(t \in d) \cdot idf(t) \cdot b_f \cdot n(q,d)) \quad (2)$$

where $tf(t \in d)$ specifies the term frequency of each term $t \in cterm$ in d, $idf(t)$ indicates the general importance of t within all documents, b_f refers to the field boost in case of an exact match of $cterm$ in the article title, and $n(q,d)$ combines Lucene-internal normalization factors. The outcome providing the better result determines the final composition of the concept term. By this, Wikipedia is acting as a concept identifier. As a result of the conceptualization step, a set of concept terms providing the basis for the automatic detection of potential dialog topics is determined. Thus, for the utterance *"Ah, then you are a fan of Bayern Munich?"* the concept terms *"fan"* and *"Bayern Munich"* are specified.

One concept term can be related to more than one topic although in various extents. Within our approach, the automatic assignment of concepts to topics is implemented by mapping all concept terms to a set of predefined Wikipedia categories. Therefore, a number of categories from Wikipedia best presenting a set of topics possibly addressed in the given dialog scenario has to be specified previously. Basically, every category contained in the Wikipedia category system can be considered to present a potential dialog topic. But it is advisable to choose those categories having a high degree of abstraction as best reflecting more general topic areas such as "Sports" or "Politics".

Subsequently, for every chosen category all subordinated Wikipedia articles are extracted, that is, all articles assigned to the considered category or to at least one of its subcategories. Afterwards, the relevant information parts are stored in a second Lucene index. More precisely, documents for every predefined Wikipedia category including field specifications about its title as well as information about the titles and textual contents of their subordinated articles are set up. Thereby, articles that are related to one predefined category several times are contained accordingly often in the category document to boost its importance within the presented topic area.

To retrieve a list of categories representing possible topics sorted in descending order according to their relatedness to the concept term $cterm$ we search the index for each category document d matching $cterm$ in a query q on the basis of the scoring formula presented in equation 2. As a result, each concept term of the present utterance is represented as a vector within a space of predefined Wikipedia categories constituting potential dialog topics. For the rest of the paper, we refer to these vectors capturing the relative importance of the dialog topics for the considered concept term as *concept topic vectors*.

Identification of Dialog Topics. As stated before, a dialog topic is established consensually from both conversation participants. That is, a single utterance does not have topics in isolation but rather provide topic suggestions [7]. Based on this idea we have to consider at least two successive utterances to define a topical intersection. Accordingly, the topic tracking process begins with the second dialog contribution.

To detect *topical overlaps* between two successive dialog contributions, we compare each of the concept topic vectors specified for one utterance with each of the concept topic vectors of the subsequent utterance separately using the *cosine similarity*. That is, we quantify the similarity between two concept terms $cterm_1$ and $cterm_2$ of successive utterances utt_1 and utt_2 on the basis of their concept topic vector representations $\overrightarrow{V}(cterm_1)$ and $\overrightarrow{V}(cterm_2)$ via

$$sim(cterm_1, cterm_2) = \frac{\overrightarrow{V}(cterm_1)\cdot \overrightarrow{V}(cterm_2)}{|\overrightarrow{V}(cterm_1)||\overrightarrow{V}(cterm_2)|} \qquad (3)$$

where $cterm_1 \in utt_1$ and $cterm_2 \in utt_2$.

If the comparing process detects a significant similarity between two concept topic vectors, that is, their similarity is higher than a given similarity threshold (currently set to 0.5), a topical overlap between utt_1 and utt_2 is identified. For every topical overlap, the involved concept topic vectors are summed up resulting in a new vector, called *dialog topic vector*. The several components in this vector provide probabilities for each predefined Wikipedia category possessing a relation to the considered concept terms. If a probability again exceeds a given probability threshold, its corresponding category constitutes the current topic of the ongoing dialog. In case the described conditions are fulfilled several times within one topic tracking process, the system is not able to determine one single Wikipedia category to be the current dialog topic but rather keeps all topic options open. Otherwise, that is if one dialog topic could be identified, the underlying dialog topic vector is included in the next identification step to keep track of this dialog topic subsequently. For this purpose, it is treated as a concept topic vector of the current utterance and is thus compared to all concept topic vectors of the following utterance to search for topical overlaps. Figure 2 graphically presents possible results of the topic tracking process for our example dialog introduced in 2.1 by means of a bar diagram. As reaching a probability ≥ 0.5 after scaling and thus exceeding the threshold represented by the horizontal line in black, the categories *"Regions"* and *"Sports"* constitute the dialog topics within this illustration.

Utterances which do not provide any concept information, like the utterance *"I know."*, have no impact on the probabilities for the several dialog topics.

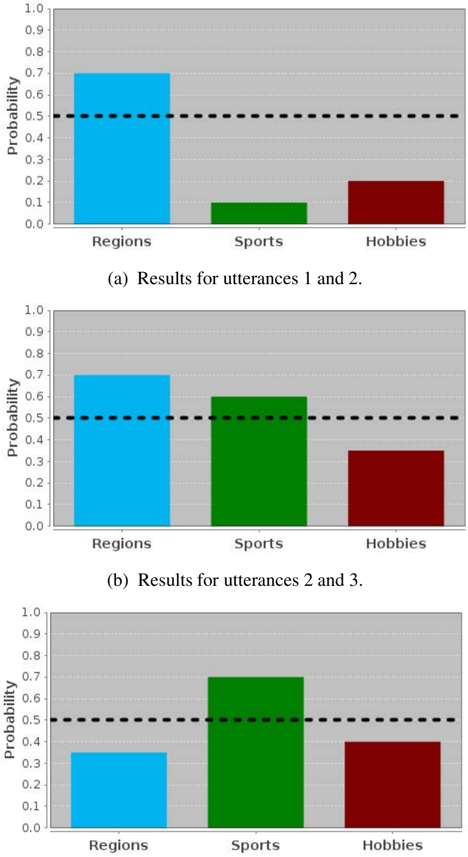
Topic Shift Detection. As mentioned before, we distinguish between a *topic leap* as described by Svennevig (1999) and a *topic drift* as introduced by Hobbs (1990). Based on this, systems are capable of detecting radical topic shifts enabling the particular conversational agents to generate an appropriate conversation behavior. According to this, the agent might refer to this topic leap via a suitable utterance such as *"What made you think of this topic?"*.

To distinguish between the two types of topic shift automatically, the transition from one dialog topic to the next is evaluated based on the outcomes of the topic tracking process. That is, if no topical overlap between the utterances utt_1 and utt_2 can be determined, the system detects a topic leap. In contrast, a topic drift is characterized in that topical overlaps to both the old and the new dialog topic exist during the topic transition as shown in Figure 2.

Topic Labeling. To be able to refer to a dialog topic later on, for example in another dialog, a descriptive topic label has to be defined. Wikipedia provides topic labels in terms of category titles. Thus, a topic can be labeled with the title of the Wikipedia category that constitutes the current dialog topic. Thereby, the labels do not have to be mentioned during dialog before as they are already existent. However, some category titles might need to be changed to more intuitive labels. The category title "Leisure", for instance, can be replaced by "Hobbies" as the latter provides a more humanlike term for a topic raised in smalltalk conversations.

4 Making Artificial Agents More Topic Aware

So far, we described how to detect topics in ongoing dialog automatically by means of collaborative knowledge provided by Wikipedia. However, to emulate humanlike



(a) Results for utterances 1 and 2.

(b) Results for utterances 2 and 3.

(c) Results for utterances 3 and 4.

Fig. 2. Bar diagrams presenting results of the topic tracking process for our example dialog (see 2.1). For clarity reasons, the display adapts automatically and only shows those bars representing active topics.

topic awareness in artificial agents our presented topic detection model needs to be embedded into the agent's underlying system architecture. More precisely, the agent's conversational behavior has to be adapted by means of the gained topic information to enable coherent dialogs between human and artificial interlocutors. In this section we propose our approach for improving the conversational abilities of the ECA Max by integrating topic information into the agent's existing dialog system.

4.1 The Conversational Agent Max

Max [15] is a virtual character acting as a conversational computer interface that allows for face-to-face dialogs with humans in German language. By means of keyboard-based, textual inputs human users are able to communicate with the agent. Max is capable of responding to these inputs with spoken language realized by a synthesized voice.

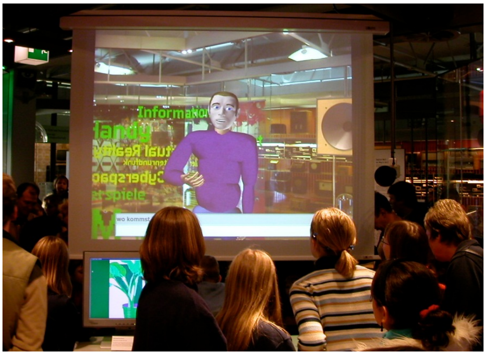


Fig. 3. Max at the Heinz Nixdorf MuseumsForum in Paderborn, Germany

Figure 3 shows Max in his current state acting as a museum guide where he provides information about the exhibition and involves human visitors in everyday small talks.

4.2 Max' Existing Dialog System

The agent's verbal communication is realized by a dialog system consisting of three modules successively processing the input of the human dialog partner. In a first step the interpreter of the dialog system determines the meaning of the user's input text. The result of this analysis is then delivered to the dialog manager. By accessing the dialog knowledge, the dialog manager chooses an according answer which is sent to behavior planning afterwards. The behavior planning component translates the answer into a multimodal utterance for the virtual character.

Both the interpretation of natural language inputs and the generation of an adequate response to the user's utterance are based on a set of rules. Thereby the interpretation is composed of two steps: First, the identification of modifiers specifying the expression type such as negation, agreement, or greeting. Second, the identification of the conversational function reflecting the pragmatic and semantic meaning of the considered utterance. These processes currently employ about 1.200 rule plans which are selected and executed via pattern matching processes. These rules in turn direct the choice of an adequate response.

Due to the rule-based input interpretation covering a broad spectrum of possible utterances and an additional, Wikipedia-based question answering component [16], the agent's system never fails in computing an appropriate reply. Hence, Max never stays speechless even if an input cannot be decoded in detail. Nevertheless, the system has not yet been able to establish coherent sequences of dialog contributions as human-like topic awareness is not accessible for the agent. The integration of our online topic detection model into the ECA's system architecture is twofold: First, we contribute to improve human-agent conversations by enabling topical dialogs between human and artificial interlocutors. Second, the existing human-machine interface provides an optimal platform for the evaluation of our approach.

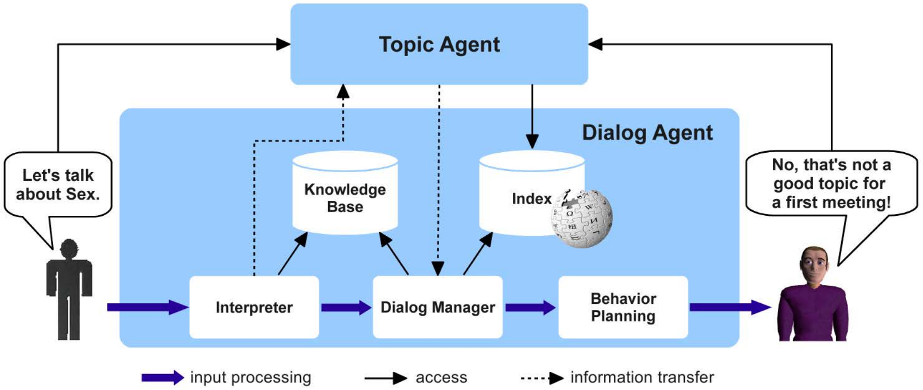


Fig. 4. Integration of our topic detection model into the existing architecture of the ECA Max

4.3 Integrating Topic Information

The complete system underlying the ECA Max is based on a multi-agent system composed of several interacting agents. The conversational behavior, for example, is realized via a dialog system in terms of an intelligent *dialog agent*. According to this, we built up a *topic agent* implementing the presented processes and integrated this agent into the existing system as shown in Figure 4.

The topic agent obtains every dialog contribution, that is the user's inputs as well as the agent's outputs, and constantly provides up-to-date information about the current topic situation of the ongoing dialog. It is directly connected to the dialog agent due to interdependencies. More precisely, the interpreter of the dialog agent sends its interpretation results to the topic agent which decides on the topical relevance of the considered utterance on the basis of the identified modifier. That is, if an utterance is specified as a greeting or farewell, the topic agent does not consider it as being topically relevant. Additionally, if one interlocutor proposes a dialog topic directly and the interpreter specifies a rejection in response to this suggestion, the topic agent again assigns the proposed topic to irrelevant topics. To give an example, if Max says *"Let's talk about music!"* and his human dialog partners answers with *"I don't want to talk about music!"*, the topic agent does not identify "Music" to be the dialog topic even if it was mentioned in two successive utterances.

The topic agent in turn sends the results of its topic detection process to the dialog manager which has an impact on the conversational behavior of Max. For this purpose, new dialog rules were defined allowing the agent to give information about the current dialog topic, to wonder about sudden topic shifts (i.e. topic leaps) and to restrain the search domain for the question answer component [16]. Moreover, the rules contained in the knowledge base triggering or processing topic suggestions are topically arranged to distinguish between their adequacies according to the given dialog setting. In the following, an example extract of the resulting rule library based on the agent architecture *JAM* [17] is given.

```
/** TOPIC: REGIONS **/
Plan {
  NAME: 'askFor-homeTown'
  GOAL: get user's home town
  BODY: <act> Where are you from? </act>
  ...}
Plan {
  NAME: 'tell-favoriteCountry'
  GOAL: tell system's favorite country
  BODY: <act> I love Portugal. </act>
  ...}

/** TOPIC: SPORTS **/
Plan {
  NAME: 'reply-likingSports'
  GOAL: tell system's interest in sports
  BODY: <act> Yes, I like soccer. </act>
  ...}

/** TOPIC: POLITICS **/
Plan {
  NAME: 'askFor-politicalAttitude'
  GOAL: get user's party affiliation
  BODY: <act> What's your preferred
        political party? </act>
  ...}
```

The topical classification of the rules allows their execution based on the dialog situation. Given a first encounter, the dialog participants would not talk about their political affiliation, for instance. Accordingly, Max avoids making use of the rules dealing with so-called *unsafe topics*. That is, he neither uses such a rule pro-actively raising a topic nor reactively to answer a user question. Regarding the latter, he rather gives an evasive answer (as shown in Figure 4).

5 The Dialog Scenario

In our scenario, a human participant has a face-to-face small talk encounter with the virtual agent Max. Thereby, the human dialog partner expresses him or herself via keyboard-based text inputs whereas the artificial interlocutor answers with spoken language based on speech synthesis. Thus, the contributions of either side exist as textual information redundantizing additional speech recognition processes. Moreover, preprocessing steps to handle incomplete and non-standard sentences are not required as typed inputs mostly consist of complete sentences containing only little abbreviations and slang expressions. However, textual inputs preclude the perception of topic ending indicators such as repetitions, pauses, laughter, etc. [18]. Thus, they can not be considered in the process of topic detection although often used in human conversation.

Enabling a coherent dialog between Max and a human user necessitates the presetting of a topical structure allowing to introduce the emulated topic awareness into the dialog and to cause a corresponding conversational behavior on the agent's part. Schneider (1988) assigns a structure to a typical small talk sequence as follows:

1. Question
2. Answer
3. Reply
4. Further turns

Furthermore, a typical small talk topically covers the immediated, external, and communication situation [11]. In their study, Endrass et al. (2011) identified a typical distribution of these topics within a dialog between Germans. Thus, Germans address less of the immediated and approximately equivalent of the external and communication situation during small talk. According to these findings, and considering the conditions arising from the fact that Max is situated in an university environment, the beginning of topical small talks with the conversational agent Max is structured as follows: In his first turn, Max asks the interlocutor for his or her subject of study as most potential dialog partners are students. Subsequently, the agent tries to find out the interlocutor's origin. If successful, Max is able to determine the interlocutor's favorite football club from this knowledge and to continue with the football topic. In case the human does not want to talk about football or sports in general, he or she has the opportunity to suggest another topic. Thus, the first dialog topics are solely initiated by Max. This is important insofar as this scenario also establishes a basis for the development of a personal memory for the agent as introduced by Mattar and Wachsmuth [19]. This requires the gathering of a lot of information concerning social categories about the human interlocutor.

5.1 Planned Evaluation

Upon successfully completing a preliminary evaluation identifying the topics of newspaper articles, which has shown an average accuracy of 61.0 [8], we plan to accomplish a more adequate evaluation considering and addressing the introduced dialog scenario. Accordingly, we searched for a corpus comprising dialog information of German small talks occurring during first encounters between two persons. The CUBE-G corpora [4] provides analyzed records of 21 first interaction scenarios each between a student and a professional actor and each lasting around five minutes. Amongst others, the dialogs were tested for the amount of topics and topic shifts which is why the corpus contains topical annotations for each recorded small talk. Thus, the CUBE-G corpus presents the optimal basis for our following evaluation.

In preparation for the planned evaluation, we already determined a list of predefined main categories that represent typical dialog topics for everyday small talks. Thereby, we omitted so-called *unsafe topics* (see Section 1) and especially focused on topics raised in the given university scenario. Table 1 shows the resulting list of main categories. Moreover, we downloaded the German database dump from May 14, 2011 and built up a Lucene index containing all information parts relevant for our purpose.

Table 1. List of predefined main categories adequate for our dialog scenario

Main Category	
Science	Economics
Family	Education
Studies	Literature
Mass media	Music
Arts	Health
Ecology	Digital media
Sports	Occupations
Fashion	Food and drink
Leisure	Transport
Intimate relationships	Regions

The next step is the preprocessing of the corpus in that incomplete sentences and expressions are completed to adapt the recorded utterances to the conditions given by the fact that human-sided utterances are based on keyboard inputs. Then, we will accomplish the evaluation by automatically identifying the dialog topics and topic shifts within the CUBE-G interactions by means of our proposed method to subsequently compare the results with the manual annotations included in the corpus. If showing promising performance, a user study evaluating the application of emulated human topic awareness in the agent Max' conversational behavior will be scheduled next.

6 Related Work

A lot of work has been carried out on offline topic identification. A prevalent model was developed in the context of the Topic Detection and Tracking (TDT) research program [20]. Within the TDT research, Allan determined five tasks (i.e., Story Segmentation, First Story Detection, Cluster Detection, Tracking, and Story Link Detection) for detecting the several topics outlined in a text-based newscast. Further offline approaches compute the coherence between documents via similarity measures (e.g., [21,22]). Others rank Wikipedia articles according to their relevance to a given text fragment, for example via text classification algorithms [13] or by simply exploiting the Wikipedia article titles and categories [23]. One recent approach uses the Wikipedia category network as a conceptual taxonomy and derives a directed acyclic graph for each document by mapping terms to a concept in the category network [24].

Approaches for the online identification of topics in natural language dialogs are rare. One work realizing a “Dynamic Topic Tracking” of natural language conversations between a human and a robot roughly adapted the five tasks from the TDT project (see above) to make the robot more situation aware in human-robot interaction [25]. Thereby the amount of topics and the according topic names are created dynamically by gathering the topic names from content words most occurring in the dialog utterances. On the contrary, existing taxonomies can serve as a source for topic labels, for example derived from the online encyclopedia Wikipedia [8,16]. Furthermore, conversation clusters visually highlight topics discussed in conversations using Explicit Semantic Analysis based on Wikipedia articles [26].

7 Conclusions and Future Work

We presented an approach for the automatic emulation of humanlike topic awareness in ongoing small talk dialogs to extend the conversational abilities of a virtual agent in human-agent interactions. More precisely, we proposed solutions for both tasks the automatic identification of dialog topics and the integration of the resulting topic information into the agent's existing system architecture. The several associated processes fulfill the requirements given by a face-to-face encounter between a human and a conversational agent and enable both a coherent and socially adequate dialog between the human and the artificial interlocutors. Thereby, we exploit Wikipedia knowledge and hence the benefits originated from collaborative work (namely the existence of information whose maintenance and expansion is carried out by numerous volunteers and the reflection of the participants' common perception of conceptual structures).

In future, we will extend our approach by detecting and linking topical affiliations to previous dialog topics to handle short *side trips* to past topics. Moreover, we will resolve ambiguities by taking into account the current dialog topic to influence the concept detection process.

Acknowledgements. This work is kindly supported by the Deutsche Forschungsgemeinschaft (DFG) in the context of the KnowCIT research project in the Center of Excellence Cognitive Interaction Technology (CITEC) at Bielefeld University. We thank Birgit Endrass and Elisabeth André from the University of Augsburg for providing parts of their CUBE-G corpus.

References

1. Breuing, A., Wachsmuth, I.: Let's talk topically with artificial agents! providing agents with humanlike topic awareness in everyday dialog situations. In: Proceedings of the 4th International Conference on Agents and Artificial Intelligence (ICAART), pp. 62–71 (2012)
2. Svennevig, J.: Getting Acquainted in Conversation. John Benjamins Publishing (1999)
3. Clark, H.H.: Using Language. Cambridge Univ. Press (1996)
4. Endrass, B., Rehm, M., André, E.: Planning small talk behavior with cultural influences for multiagent systems. Computer Speech and Language 25, 158–174 (2011)
5. Cassell, J., Bickmore, T., Campbell, L., Vilhjálmsson, H., Yan, H.: Human conversation as a system framework: Designing embodied conversational agents. In: Cassell, J., Sullivan, J., Churchill, E. (eds.) Embodied Conversational Agents, pp. 29–63. MIT Press (2000)
6. Jurafsky, D., Martin, J.H.: Speech and Language Processing. Pearson Prentice Hall (2009)
7. Schank, R.C.: Rules and topics in conversation. Cognitive Science 1, 421–441 (1977)
8. Breuing, A., Waltinger, U., Wachsmuth, I.: Harvesting Wikipedia knowledge to identify topics in ongoing natural language dialogs. In: Proceedings of the 2011 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, pp. 445–450. IEEE Press (2011)
9. Bublitz, W.: Topical coherence in spoken discourse. Studia Anglica Posnaniensia 22, 31–51 (1989)
10. Hobbs, J.: Topic drift. In: Dorval, B. (ed.) Conversational Organization and Its Development, pp. 3–22. Ablex Publishing (1990)
11. Schneider, K.P.: Small Talk: Analysing Phatic Disourse. Hitzeroth (1988)

12. Toutanova, K., Manning, C.D.: Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 63–70 (2000)
13. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In: Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI 2007 (2007)
14. McCandless, M., Hatcher, E., Gospodnetić, O.: Lucene in Action, 2nd edn. Manning (2010)
15. Kopp, S., Gesellensetter, L., Krämer, N.C., Wachsmuth, I.: A Conversational Agent as Museum Guide – Design and Evaluation of a Real-World Application. In: Panayiotopoulos, T., Gratch, J., Aylett, R.S., Ballin, D., Olivier, P., Rist, T. (eds.) IVA 2005. LNCS (LNAI), vol. 3661, pp. 329–343. Springer, Heidelberg (2005)
16. Waltinger, U., Breuing, A., Wachsmuth, I.: Interfacing virtual agents with collaborative knowledge: Open domain question answering using Wikipedia-based topic models. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2011), pp. 1896–1902. AAAI Press (2011)
17. Huber, M.J.: JAM: A BDI-theoretic mobile agent architecture. In: Proceedings of the Third International Conference on Autonomous Agents (Agents 1999), pp. 236–243 (1999)
18. Howe, M.: Collaboration on topic change in conversation. Kansas Working Papers in Linguistics 16, 1–14 (1991)
19. Mattar, N., Wachsmuth, I.: Who are you? On the acquisition of information about people for an agent that remembers. In: Proceedings of the 4th International Conference on Agents and Artificial Intelligence, pp. 98–105 (2012)
20. Allan, J.: Topic Detection and Tracking: Event-based Information Organization, pp. 1–16. Kluwer Academic Publishers (2002)
21. Makkonen, J., Ahonen-Myka, H., Salmenkivi, M.: Simple semantics in topic detection and tracking. Information Retrieval 7, 347–368 (2004)
22. Zhang, X., Wang, T.: Topic tracking with dynamic topic model and topic-based weighting method. Journal of Software 5, 482–489 (2010)
23. Schönhofen, P.: Identifying document topics using the Wikipedia category network. In: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2006 (2006)
24. Chahine, C.A., Chaignaud, N., Kotowicz, J.P., Pécuchet, J.P.: Conceptual indexing of documents using Wikipedia. In: Proceedings of the 2011 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, pp. 195–202 (2011)
25. Maas, J.F., Spexard, T., Fritsch, J., Wrede, B., Sagerer, G.: BIRON, what's the topic? A multi-modal topic tracker for improved human-robot interaction. In: Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication (ROMAN), pp. 26–32 (2006)
26. Bergstrom, T., Karahalios, K.: Conversation clusters: Grouping conversation topics through human-computer dialog. In: Proceedings of the International Conference on Human Factors in Computing Systems, CHI 2009 (2009)

Towards Semantic Resources in the Cloud

Salvatore F. Pileggi and Carlos Fernandez-Llatas

ITACA-TSB, Universidad Politècnica de Valencia,
Camini de Vera S/N, 46022, Valencia, Spain
salpi@upvnet.upv.es, cfllatas@itaca.upv.es

Abstract. During the past years, the cloud vision at distributed systems progressively became the new trend for the next generation platforms. The advance in the technology, both with the broadband availability and the explosion of mobile computing, make the massive migration to cloud solution next to be a fact. The Cloud model assures a new technologic and business environment for services and applications where competitiveness, scalability and sustainability converge. On the other hand, next generation applications have to be able to pervasively meet the needs and requirements deeply different among them. Applications involving complex virtual organizations require a higher level of flexibility. An effective approach is based on the convergence between migration and virtualization. The resource-centric model assumes file systems, DBs, services and any other class of resources available in the "cloud" as Virtual Resources. These heterogeneous resources can be managed in a unique virtual context regardless by the infrastructures on which they are deployed. Semantics play a critical role in order to assure advanced and open solutions in a technologic context featured by a fundamental lack of standardization.

Keywords: Cloud Computing, Semantic Technologies, Virtual Organization, Virtualization.

1 Introduction

The popularity of Cloud technologies is constantly increasing as well as the interest on this emerging market from specialized companies that are already able to offer several solutions based on different models such as IaaS, PaaS and SaaS [1].

Even if there are several concerns from involved customers and stakeholders and open issues (e.g. privacy [2], security [3] and standardization [4]) concerning the massive migration to the cloud, both private and enterprise cloud solutions are unanimously considered the "future" of the computation [5].

In practice, the cloud approach is actually referred as the most competitive, scalable and sustainable solution on the market under the always more realistic conditions of constantly decreasing bandwidth price and of always connected users. An exhaustive analysis of technical [1] and business [6] aspects of cloud solutions is out of paper scope.

This business scenario could quickly change in the next future if the cloud will not provide high flexible solutions able to meet the needs of complex Virtual Organizations

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 407–416, 2013.

(VOs). Analyzing VOs, it is evident that not all resources normally involved are suitable for the migration to the Cloud. The level of flexibility of cloud solutions could be further increased if the platforms are the result of the convergence between migration and virtualization. The key idea is that resources deployed using cloud infrastructures (migrated resources) and resources deployed according to conventional solution could be merged in a unique virtual environment.

This paper proposes a resource-centric model for cloud infrastructure in which virtual resources are provided with a semantic description/specification able to assure a potential high level of interoperability among platforms as well as a set of facilities for the integration of pre-existent or new resources. As detailed in the following sections, semantics play a critical role in the proposed model especially considering the fundamental lack of standardization the cloud is experimenting [4].

As any cloud model, it is potentially independent from any application domain even if, inevitably, domain-specific semantic representations could be required in certain contexts/applications.

A potential application domain for the model is the Spanish health system. In Spain, the National Health System follows a decentralized model where each autonomous community manages all the centres, services and establishments of the Community Councils, Town Councils and any other intra-regional governments. An autonomous community is the first-level political division of the Kingdom of Spain, established in accordance with the current Spanish Constitution. Each autonomous community has many hospital systems of very different natures, being independent of each other.

When a patient must go to different health centres should generally answer the same questions in order to open his/her medical records in each of them. This generates an overall lack of coordination regarding the interoperability between systems, redundant data, basic services, etc.

There is a current trend, which aims to unify those different systems to autonomous community level. At the moment, interconnection between all hospital systems is unapproachable, being applied to European context.

With this purpose, the proposed model aims to provide interoperability through cloud-based environments, giving support and tools for the virtualization, migration of already existing systems formed by data, apps and/or infrastructure and creation of new generation services, in order to make available basic and composite services to consumers (patients, services,…) and for development and management of customized e-health services to all users.

The core part of the paper is structured in 3 main sections: the section 2 is the natural extension of the introduction (it provides a detailed description of approach and scope); in the section 3 the model is deeply described; finally, in the section 4 a brief analysis to next generation resources is provided.

2 Approach and Scope

The scope of enabling semantic resources in the cloud is the interconnection of heterogeneous systems providing the coexistence of different kinds of heterogeneous resources (Figure 1):

- *Internal Resources*: they are the result of the common migration process to the cloud. Resources are hosted by internal infrastructures and they are available as virtual resource in the platform.
- *External Resources*: they are hosted by external infrastructures but they are pervasively available into the platform as virtual resources (Virtualization).
- *Next Generation Resources*: can be designed and implemented directly over the virtualized layer provided by the platform.

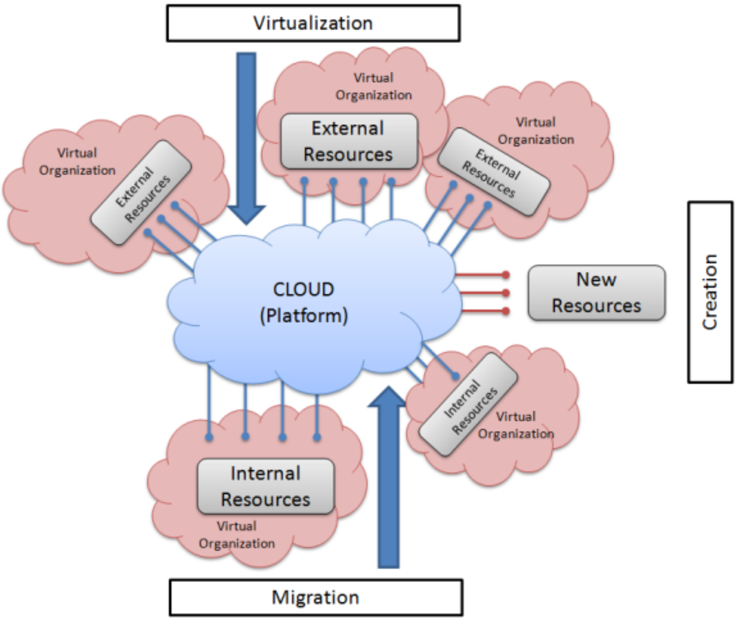


Fig. 1. Interconnecting resources in the Cloud

The platform model assures a flexible and shared infrastructure mainly featured by the following points:

- It enables *ecosystems* among heterogeneous systems.
- It provides a pervasive *virtual environment*: resources are managed at virtual levels assuring high-interoperable capabilities.
- *Cloud Approach*: scalable, competitive and sustainable solutions.
- *Open Semantic Support*: the core semantic support can be integrated/extended with domain-specific interoperability capabilities (e.g. health/medical).
- *Cross-domain Platform*: the core infrastructure of the platform is not domain-specific; the potential application range increases with the expressivity of the semantics.

The diagram represented in the figure 2 shows the platform conceptualization. The free software IHMC CmapTools [7] is adopted for specifying this compact model as a concept map. IHMC CmapTools proposes an approach for knowledge representation similar to semantic networks that build semantic relations among concepts through a directed or undirected graph consisting of vertices, which represent concepts, and edges.

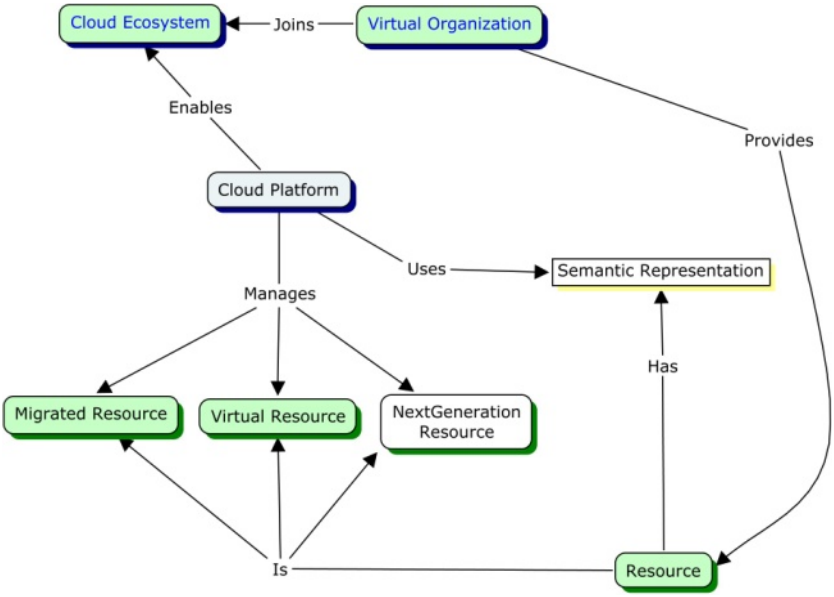


Fig. 2. Platform conceptualization

As showed, pre-existing VOs join the common cloud ecosystem enabled by the platform. Each VO provides a set of resources. These resources can be public (available in the ecosystem for any other VO), protected (available just for authorized actors) or private (available only inside of owner VO or under payment). The platform is able to manage any kind of resource regardless by the infrastructures on which they are deployed. This interoperable layer is assured by the semantic representation of resource.

2.1 A Practical Use Case: Health Systems Interconnection

Several medical systems (e.g. ORION and IANUS) are currently coexisting in Spain providing similar data and services. They are solutions developed in different times in order to meet different requirements and needs from different end-users (e.g. hospitals) that are progressively converging.

Integrated services as well as the need of a stronger level of collaboration require a high level of interoperability. This requirement is completely missed in current systems that are designed to work according to stand-alone behaviors.

The interoperability among health systems is normally solved through ad-hoc solutions (e.g. an interoperable layer between ORION and IANUS). This class of solution is expensive and has several limitations in terms of flexibility because it does not solve a generic problem but just a concrete/local problem. Details about will be provided in the section 3.1.

The health systems interconnection is probably an extreme case from several points of view. But it is a significant example of the impact that virtual (or real) organizations have or can have on the development of systems and platforms operating in the real world. Business scenarios, constraints and restrictions advise an increased level of flexibility in order to assure sustainable and effective environments.

3 Platform Model

The convergence between cloud-based solution and virtualization is not an absolute novelty. An exhaustive analysis [8,9] is out of paper scope but, recently, this approach was used in order to reach different goals in the context of different domains.

The last generation of virtual platforms [10], virtual organizations [11], virtual services [12], techniques for virtual resources management [13]/optimization [14] and virtual infrastructure [15] are, implicitly or explicitly, referred to the cloud model.

Also a semantic specification of resources is a well known topic for both general purpose and specific (e.g. industrial resources [16]) purpose.

The proposed model is composed of three converging perspectives:

- *Interoperability model* (Section 3.1)
- *Business model* (Section 3.2)
- *Technical perspective* (Section 3.3)

3.1 Interoperability Model: Vertical Approach

One of the critical and key issues for the improvement of the current interoperability model is the conceptual evolving from a “horizontal” to a “vertical” approach.

Actually medical systems are logically part of virtual organizations characterized by different complexity in terms of structure and distribution. Each virtual organization has its own technological environment.

During the last few years, a progressive convergence among these environments was aimed (see introduction). The problem is normally approached trying to provide added (or improved) capabilities among existent systems (Figure 3). This “direct” solution is effective and efficient but it is just a local solution: the “integration” of a new system (or resource) implies the need of a “new” solution.

Furthermore, if a new system/resource is integrated in the ecosystem and it has to be interoperable with the existent ones, an ad-hoc component (proxy) that assures the interoperability has to be provided for interfacing each existent system/resource.

This last situation is expressed by (1) where n is the number of independent systems/resources and k is the number of proxies. As showed, if a new resource or system is integrated, a full-interoperable solution implies the deployment of O(n) proxies.

$$\begin{gathered} k(i) = k(i-1) + n(i-1) \rightarrow O(n-1) = O(n) \\ n>2 \end{gathered} \tag{1}$$

The virtualization of resource enables a model of interoperability based on a vertical approach (Figure 3): resources are available at virtual level and the interoperability among systems is approached at this abstracted level.

A solution based on the vertical approach for the interoperability is intrinsically simplest because the integration of a new system/resource just implies the interfacing with the abstract layer (2).

$$\begin{gathered} k(i) = k(i-1) + 1 \rightarrow O(1) \\ n>2 \end{gathered} \tag{2}$$

In practice the vertical approach can be assured according to two different approaches:

1) Standards. Solutions based on shared layers that impose resources to be described according to a well defined set of standards. This is the simplest solution by a technical and conceptual point of view but, unfortunately, is not always easy to be applied in the context of complex virtual organization. This is mainly because standards are hard to be imposed and the data/knowledge from systems can be hard to be converged in well defined standards even considering domain-specific solutions.

2) Open models. Dynamic solutions based on open models (e.g. semantics). Semantic representations guarantee the definition of local knowledge environments that can be centrally managed without the need to share standards. Furthermore, resources could interact among them interchanging semantic data. A completely open model is complex to be proposed and managed. As explained in the section 4, realistic solutions can be designed according to hybrid approaches based on core ontologies that can be extended and/or particularized.

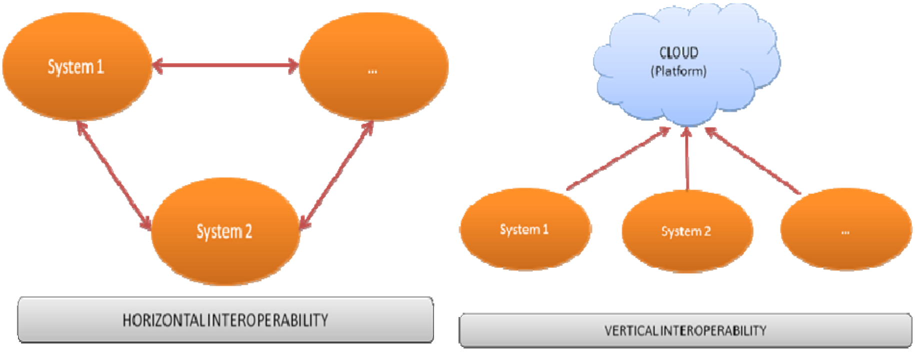


Fig. 3. Horizontal (up) vs vertical (down) approach for interoperability

3.2 Business Model: Merging Migration and Virtualization

Merging migrated and virtualized resources provides a high level of flexibility respect to both technologic and business perspectives. Migration assures a scalable environment, the potential reduction of maintenance costs, as well as the other advantages typical of the cloud approach.

Virtualization allows a further degree of flexibility for resources that owners cannot migrate or do not want to migrate: resources are available into the ecosystems but they are not migrated to the cloud. Motivations for preferring a virtualized resource can be related to law restrictions, business constraints or any other real situation that does not match a full cloud approach.

Regardless from the deployment aspects, a platform that allows the enablement and integration of heterogeneous resources significantly improves the business opportunities and capabilities around shared resources. This is mainly because the virtual resources repository provided by the platform allows a central understanding and management of distributed and heterogeneous environments. The exploitation model of available resources is conceptually similar to services exploitation model. An exhaustive analysis is out of paper scope.

3.3 Technical Overview: Functional and Semantic Support

Platform designed according to the proposed approach should include at least a set of core functionalities, as well as a full semantic support. A reference model can be logically structured as in the follow (Figure 4):

- *CORE Infrastructure/Functional support*: infrastructures and any other functional support to the ecosystem.
- *Management support*: set of functionalities for the management of the platform.
- *Semantic support*: models for knowledge representation.

An exhaustive overview at the architecture design (as well as a detailed analysis of functional layers) is out of the paper scope. The most interesting functional component is probably the *Enabler*.

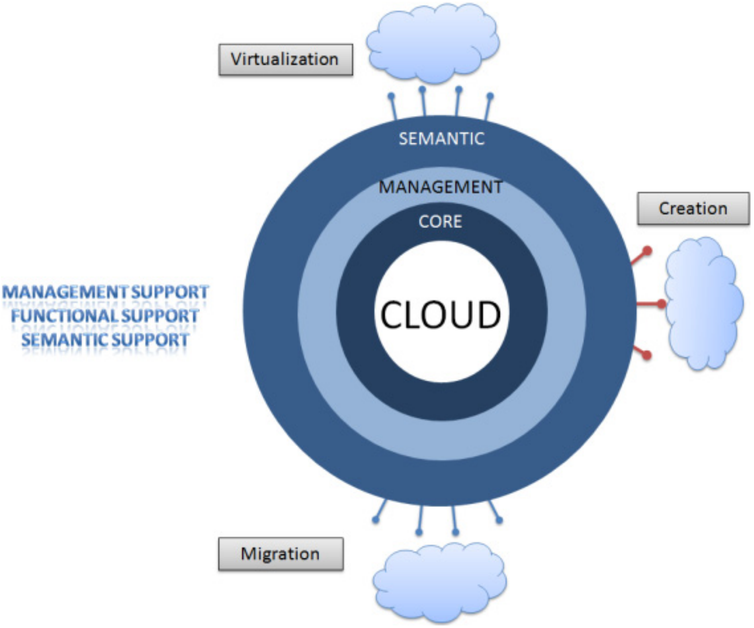


Fig. 4. Technical overview at the platform

Enablers have the critical role of allowing the access to the resources and/or to the modules information. In practical cases, a refinement of the enablers could be required: the information related to a single module could not be atomic and so the enabler could allow the access to a part of the fields, denying the access to other fields; from a resource point of view, also resources could be not managed as atomic components since complex access and consuming policies (similar to services) could be applied. A concrete combination of the enablers defines, at the same time, the rights of a concrete application on the resources and on the information, the role of the application user as well as the privacy contract between the resource owner and the application.

Furthermore, in order to assure a realistic exploitation plan a support for developers is required: a set of APIs and interfaces that support the developer to migrate and virtualize existing resources as well as the design and implementation of new resources.

The platform is evidently the result of the convergence among cloud technologies, virtualization techniques and semantics.

4 Next Generation Resources

A pervasive virtual environment designed according to a cloud approach can provide a solid support for the development of a new generation of resources (e.g. services and applications).

These resources can be developed directly on the top of the virtual layer provided by platforms supporting a high level of abstraction (e.g. role-driven development).

As introduced in the section 3.1, the key issue is the efficient and effective application of open models for the knowledge specification and representation.

The use of "standard" ontologies could be the most immediate solution: rich data models could be enough expressive to represent the knowledge as well as to assure inferred knowledge and an interesting set of interoperability capabilities. But it could limit the advantages and benefits provided by open solutions as well as the problems related to the knowledge convergence could not be solved or skipped.

On the other hand, a completely open model that assumes each local system/resource described according its own ontologies could be hard to be applied in real systems. Typical problems in multi-ontology computation (e.g. correctness and ambiguities) both with the objective difficulty to provide a centralized management for resources advise more realistic approaches.

The current idea is the use of shared vocabularies. These vocabularies should provide the basic concepts making possible the definition of independent local knowledge environments that can be globally linked and processed. In practice, shared concepts have to be used in order to link local ontologies to the platform. Further concepts, as well as rules and relations among them, can be provided by local ontologies. This approach is equivalent to object extension in object-oriented environments.

At the moment of designing a new resource, developers could have a full functional support provided by the platform and a dynamic semantic support. The developer can choose the deployment model (migration or virtualization) that better matches the business needs, link the resource to the platforms through concepts from the shared vocabulary and make available the knowledge required (local ontologies). Further advantages are provided at the moment to design resources that assume the coordinated/uncoordinated use of other resources that can be directly managed at high level.

5 Conclusions

The convergence between cloud and virtualized solutions in a semantic context provides improved interoperability capabilities as well as a competitive environment for resources integration.

The flexibility assured by open models for the knowledge definition and representation could play a key role in several concrete environments (e.g. Spanish health system) involving complex virtual organizations.

The power of integrating existent resources (as well as the design of new ones) directly on the top of an abstracted layer provides a new vision at the cloud and its exploitation model.

Finally, a semantic layer able to link resources to the global environment (platform) and to support, at the same time, local knowledge representations could provide a dynamic support for the effective convergence of dynamic resources in the cloud.

References

1. Zhang, S., Zhang, S., Chen, X., Huo, X.: Cloud Computing Research and Development Trend. In: Second International Conference on Future Networks, ICFN 2010, January 22-24 (2010)
2. Esteves, R.M., Rong, C.: Social Impact of Privacy in Cloud Computing. In: 2010 IEEE Second International Conference on Cloud Computing Technology and Science, CloudCom, November 30-December 3 (2010)
3. Ramgovind, S., Eloff, M.M., Smith, E.: The management of security in Cloud computing. In: Information Security for South Africa (ISSA), August 2-4, pp. 1–7 (2010)
4. Ortiz Jr., S.: The Problem with Cloud-Computing Standardization. Computer, 13–16 (July 2011)
5. Kshetri, N.: Cloud Computing in Developing Economies. Computer 43(10), 47–55 (2010)
6. Chang, V., Bacigalupo, D., Wills, G., De Roure, D.: A Categorisation of Cloud Computing Business Models. In: 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), May 17-20 (2010)
7. IHMC CmapTools, online at: http://cmap.ihmc.us/ (accessed on September 24, 2010)

8. Dong, H., Hao, Q., Zhang, T., Zhang, B.: Formal Discussion on Relationship between Virtualization and Cloud Computing. In: 2010 International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT (2010)
9. Gmach, D., Rolia, J., Cherkasova, L.: Resource and virtualization costs up in the cloud: Models and design choices. In: 2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks, DSN (2011)
10. Siddhisena, B., Warusawithana, L., Mendis, M.: Next generation multi-tenant virtualization cloud computing platform. In: 2011 13th International Conference on Advanced Communication Technology, ICACT (2011)
11. Li, J., Li, B., Du, Z., Meng, L.: CloudVO: Building a Secure Virtual Organization for Multiple Clouds Collaboration. In: 2010 11th ACIS International Conference on Software Engineering Artificial Intelligence Networking and Parallel/Distributed Computing, SNPD (2010)
12. Fu, J., Hao, W., Tu, M., Ma, B., Baldwin, J., Bastani, F.B.: Virtual Services in Cloud Computing. In: 2010 6th World Congress on Services, SERVICES-1 (2010)
13. Van Nguyen, H., Tran, F.D., Menaud, J.-M.: SLA-Aware Virtual Resource Management for Cloud Infrastructures. In: Ninth IEEE International Conference on Computer and Information Technology, CIT 2009 (2009)
14. Khatua, S., Ghosh, A., Mukherjee, N.: Optimizing the utilization of virtual resources in Cloud environment. In: 2010 IEEE International Conference on Virtual Environments Human-Computer Interfaces and Measurement Systems, VECIMS (2010)
15. Keller, E., Szefer, J., Rexford, J., Lee, R.B.: NoHype: Virtualized Cloud Infrastructure without the Virtualization. In: Proceedings of the 37th Annual International Symposium on Computer Architecture, ISCA 2010 (2010)
16. Khriyenko, O., Terziyan, V.: OntoSmartResource: an industrial resource generation in semantic Web. In: 2004 2nd IEEE International Conference on Industrial Informatics, INDIN 2004, June 26 (2004)

Researching Nonverbal Communication Strategies in Human-Robot Interaction

Hirotaka Osawa and Michita Imai

Faculty of Science and Technology, Keio University,
3-14-1, Hiyoshi, Kohoku-ku, Yokohama, Japan
{osawa,michita}@ayu.ics.keio.ac.jp

Abstract. We propose an alternative approach to find each robot's unique communication strategies. In this approach, the human manipulator behaves as if she/he becomes the robot and finds the optimal communication strategies using attachable and detachable robot's shapes and modalities. We implement the system including a reconfigurable body robot, an easier manipulation system, and a recording system to evaluate the validity of our method. We evaluate a block-assembling task by the system by turning on and off the modality of the robot's head. Subsequently, the robot's motion during player's motion significantly increases whereas the ratio of confirmatory behavior significantly decreases in the head-fixed design. In this case, the robot leads the users and the user follows the robot as in the turn-taking communication style of the Head-free condition.

Keywords: Design Methodology, Human Robot Interaction, Human Interface.

1 Introduction

Nowadays, robots having various kinds of shapes and modalities can support our lives in many ways. In this paper, we define shape as the appearance of the robot and modality as the possible observation and behavior of the robot. There are still questions about what kind of interaction is required for each robot shape and modality [1]; [2].

Previous studies have designed and implemented the shape and modalities of robots according to human-human interaction. There are many studies that referred to humanlike modalities in robots, such as gesture [3], manner [4], timing [5], and bipedal walking [6]. This process is conducted as shown in the two figures on the left side of Fig. 2. First, the researchers extract a psychological finding from human-human interaction and create an interaction model from it. Second, they implement the model to a humanlike robot. Third, they conduct an interaction between a human and a humanlike robot and confirm that the robot can interact as the proposed model. Such a design method is widely used in human-robot interaction (HRI) studies because of the following reasons. First, the researchers can base the study on psychological findings that have been already investigated. Next, it is easy to compare the results and the goals. The above-mentioned reasons and method allow the researchers to incorporate the contributions of previous studies.

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 417–432, 2013.

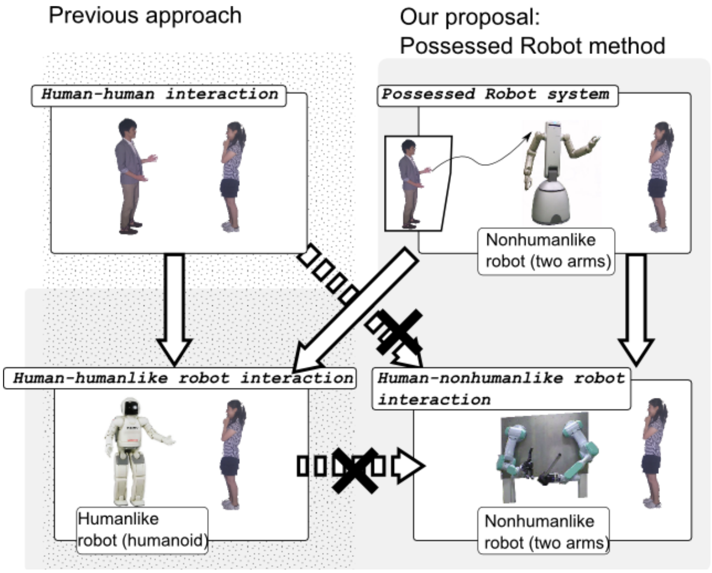

Previous approach
Our proposal:
Possessed Robot method
Human-human interaction
Possessed Robot system
Nonhumanlike robot (two arms)
Human-humanlike robot interaction
Human-nonhumanlike robot interaction
Humanlike robot (humanoid)
Nonhumanlike robot (two arms)

Fig. 1. Difference between the previous HRI approach and our proposal

However, we cannot find the specific behaviors of a robot that are not related to human shape and modalities by referring to existing findings in human-human interaction. With the above process, we may miss the most appropriate communication strategies for the robot if the robot and the human modalities are not the same. We call this kind of robot a "nonhumanlike" robot. In this paper, we use the word "nonhumanlikeness" to describe the lack of humanlike social appearance, such as humanlike head and arms. Detailed examples are shown in Fig. 2 [3]; [6]; [7]; [8]. Several HRI studies about less humanlike robots suggest that imitating humans is not the only approach to designing a robot. Sometimes, we use different communication strategies for nonhumanlike agents. One of the best examples is the human-pet interaction. Our interaction style with pets is different from our style in human-human interaction and human-tool interaction. Robots have both aspect of tools and pets. They generate different types of interaction to users using different shapes and modalities from that of humans, even if the shapes and modalities are nonhumanlike. For example, Mu, eMuu, and Social Trash Box extracted the essence of human interaction and created an abstracted relationship to humans different from human-human interaction [9]; [10]; [11]. Animal robots like Paro and AIBO result in specific interaction experiences by merging animal-like features with the original robot's modalities [12]; [13]. Training with additional humanlike features of an object allows us to use a communication strategy by merging the original features of the object and humanlike features [14]; [15]. However, there is no design method to find original communication strategies for robots except the analogical method (i.e., deriving metaphors and abstractions from existing design). This shortcoming of the previous approaches prevents us from building a robot design on human-nonhumanlike robot interaction (right bottom area on Fig. 1) because we cannot directly apply human-human interaction findings to nonhumanlike robots.

We propose the alternative method to find a specific communication strategy for a robot that can consider its own shape and modalities. In this approach, one person "possesses a robot," and behaves as if she/he is the robot while interacting with another person. This trial-and-error interaction process between two persons reveals original communication strategies that are reasonable and specific to each robot's shape and modalities. Our approach is applicable to both humanlike and nonhuman-like robots, shown in Fig. 2. If the method is applied to a two-arm and headless robot, the results are also applicable to another robot that has the same design (shown in the right side of Fig. 2).

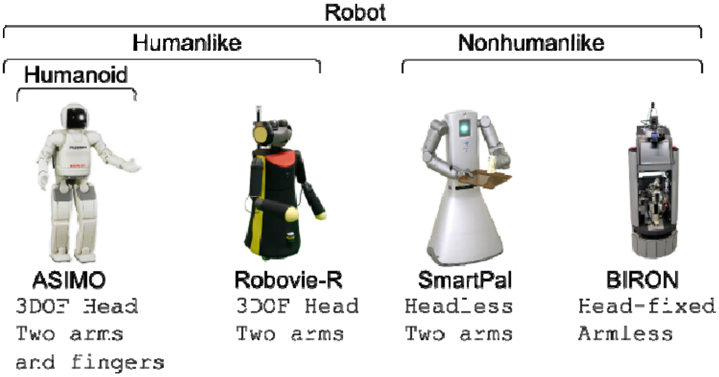


Fig. 2. Different styles of robots: their shapes and modalities

In this paper, we implemented the Possessed Robot demonstrative System (PoRoS) to validate our approach. PoRoS allows the user to possess the robot by converting the user's behavior to the robot's output and by converting the robot's input to the user's input. We evaluated our proposal with demonstrative tasks to instruct a user on how to assemble a building from wooden blocks using a robot by changing the robot head modality. A humanlike robot with head modality resembles human modalities and allows us to use conventional communication strategies, such as nodding and shaking motions. However, a humanlike robot without head modality requires different communication strategies that cannot be achieved with the existing human communication theory. Headless or head-fixed robots, such as BIRON and SmartPal, are also popular [7]; [8]. The demonstrative task also answers what kind of communication strategies are more appropriate to the commonly used headless robots.

The following sections are organized as follows. Section 2 explains the differences between related methods and studies (Wizard of Oz, teleoperation robot, and marionette system) and the Possessed Robot method. Section 3 explains the design process of the Possessed Robot method, and section 4 explains in detail the implementation of PoRoS (Possessed Robot System) for realizing the Possessed Robot method. In section 5, we explain the evaluation of PoRoS and the results are presented in section 6. In sections 7 and 8, we discuss the results and the conclusions, respectively.

2 Related Studies

In spite of differences in policy, there are several similarities between previous approaches and ours. In this section, we compare our work to related studies and clarify our contribution.

2.1 Wizard of Oz

The Wizard of Oz (WoZ) method is used mainly in evaluating computer interfaces [16]. This method uses human manipulator as sensors to avoid unessential errors from the evaluation. The WoZ experiment method is also widely used in the field of HRI. Steinfeld et al. inferred several consequential evaluation methods (called Oz of Wizard) from WoZ for evaluating robots behavior [17].

WoZ uses a human manipulator as part of the experimental interface system instead of being autonomous. The manipulator behaves as the decision maker in the system and selects the system behavior from a determined list. The role of the human manipulator in WoZ is restricted to replace sensor actions. We extend the notion of WoZ in the field of robotics using attachable and detachable robotic devices and sensors. The entire robot input and output are directly connected to the manipulator, and the manipulator behaves as an intelligent computer in finding the most optimal communication strategies for each task using the specific robot shape and modalities.

2.2 Teleoperation Robot

Teleoperation robot studies also use manipulated robots. The robot design is sometimes verified and analyzed by recorded results. Kuzuoka et al. discussed the optimal instructions in teleoperation [18]. However, teleoperation studies themselves are not designed to find the optimal communication strategies in autonomous robotic systems. If the system behaves autonomously, it is not teleoperation anymore.

Several research groups proposed to use teleoperation to complement an autonomous robot. Glas et al. proposed to use a human manipulator to guide the robot [19]. In their approach, the robot behavior is replaced by the human manipulator if the task is hard for the robot to solve. Thus, a human manipulator can temporarily possess the robot. However, their study only focused on improving the task performance in a real world human-robot interaction. This approach did not focus on feedback to optimize communication strategies. They also hypothesized that the robot might use humanlike modalities in the future. Other robot possibilities are also not well discussed in their paper.

2.3 Marionette and Digital Puppetry

Marionette is a well-known art for making puppets behave lifelike (they are sometimes humanlike and sometimes nonhumanlike). Currently, the possibility of interactive marionettes is accelerated by technology. They are called Digital Puppetries. This kind of system allows us to control humanlike and nonhumanlike robots [20]. Turtle Talk with Crush is the most successful marionette in the commercial field [21]. It is a screen agent that interactively changes its face and behavior according to people's responses.

However, these studies are specialized to each robot's shape and modalities. Manipulation requires not a small amount of training time although interface is supported by today's technologies. This marionette system is not appropriate for the trial-and-error approach that required in our method.

3 Designing the Possessed Robot Method

Possessed Robot method is a design method conducted by two participants. One participant possesses a robot and behaves as if she/he is the robot. Another participant interacting with robot.

Based on the differences to previous studies mentioned in above section, we estimate that the following three sub goals are required to perform the Possessed Robot method. First, the Possessed Robot method requires a reconfigurable robot body to examine all kinds of robot shapes and modalities. Second, the manipulation method must be easy for the human manipulator to allow frequent trial-and-error efforts. Third, the system requires recording the interaction between the robot and the human for later analysis.

The entire design process is described below:

- Select the robot input and output, and configure the robot shape and modalities.
- Assign two persons as the manipulator (who possesses the robot) and the player (who follows the robot).
- Connect the robot input and output to the manipulator. All connections are required to be understood and controlled by humans.
- Two persons interact via the robot and conduct a task cooperatively. They repeatedly try to interact and gradually find the most optimal communication strategies for the task. The system records the entire interaction.
- The evaluators analyze the result of the interaction and the kind of modalities, which are the most and least required. We also compare the results with the human-human interaction findings, which is the original interaction setup for the robot.

This process brushes up the robot design. If we require a more detailed analysis, we can also select more optimal shapes and modalities with the results from process 5, and repeat the entire process.

4 System

We implemented PoRoS (Possessed Robot demonstrative System) to estimate the validity of our process. We used a reconfigurable robot, a monitoring device to capture movement, and a recording system to solve the sub goals mentioned in the previous section.

4.1 A Reconfigurable Robot That Allows Us to Use Variable Shape and Modalities

In the Possessed Robot method, we can evaluate not only the humanlike robot shape and modalities but also any kind of shapes and modalities. For evaluating the Possessed Robot method clearly and rapidly, we created a robot kit that has separate body parts and allows variable shapes and modalities. The kit includes three axis heads and two four-axis arms. Each head has three motors. Each arm has two motors on the root of the device to achieve movements toward the pitch and yaw directions of the arm. It has also two motors on the tip to achieve movement toward the pitch and roll directions of the hand.

These devices are attachable and detachable by Velcro tapes. Each head and arm are wired and connected to a microcomputer, and can be separately turned on and off. The total axes of the kit are sufficient to reproduce normal humanlike robots. If you want to turn off the modality of the head of the robot, just turn off the switch and the robot stops controlling the head. If you want a different humanlike robot shape, you can detach each part and attach it on a different position. In the experiment, we assigned each part as in Fig. 3 left and compared the communication strategies of the humanlike robot by turning on and off the head of the robot.

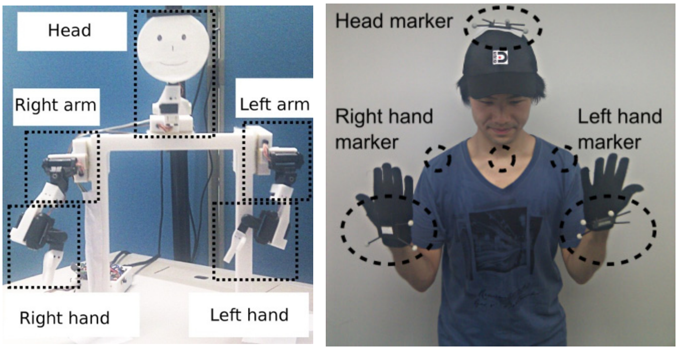


Fig. 3. Implemented reconfigurable robot on PoRoS system and motion capture markers on a participant

4.2 Monitoring Device Using the Motion Capture System

To use a human as the controller of the robot, we need to monitor the behavior of the human manipulator and feedback the robot with it. We used a motion-capturing system for feedback from the human manipulator because it is easy to understand how to move a robot. In this system, we used seven motion-capturing cameras (OptiTrack s250e [22]) for tracing the human head and hands. Each human body part is captured and converted to robot body movement as described below:

- Head: The system extracts three angles (yaw, pitch, and roll) of the head and assigns them to the robot's head movement.
- Arm: The system calculated the robot's arm angles (yaw and pitch) by a vector from the head position to the hand position.

- Hand: The system calculates the robot's hand angles (pitch and roll) by directions of the user's head.

Each marker is attached to the human body as in Fig. 3 right. Head markers are attached on the top of the manipulator's head. Hand markers are attached on the back of the manipulator's hands.

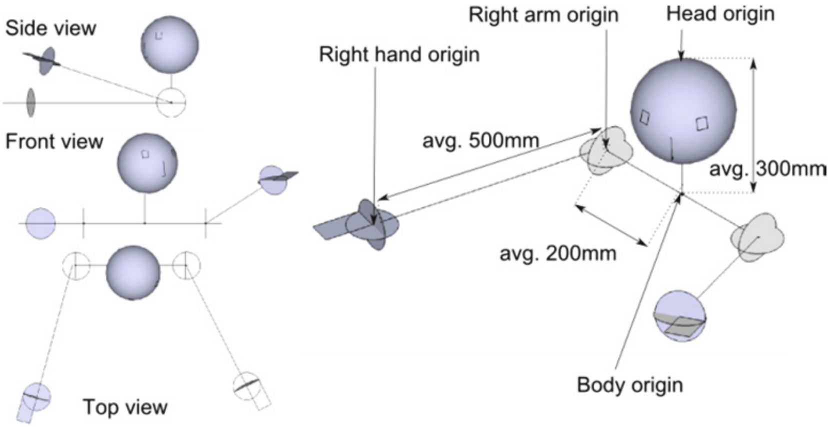


Fig. 4. Calculation method for the position of each part

All origins are calculated as in Fig. 4. First, the system calculates the centre of the human body using the top of the head. The average position of the centre of the body is 300 mm below the head top. Second, the system calculates the origins of the right and left arm from the centre of the body. Each origin is on average 200 mm from the centre of the body. We can estimate that the origins of the arms are stable because the manipulator stands in front of the video and does not change her/his shoulder angle. Third, we calculate the arms' vectors from each angle and arm's length (average 500 mm). Last, we assign the hands' directions toward the pitch and roll axis of the robot's hands.

4.3 System Connections

All modules are connected as in Fig. 5. In PoRoS, the input data to the human manipulator is the video image and the output data from the human manipulator are the motion-capturing data and angles of each motor. The latency from the robot to the user is below 200 ms and this delay does not cause any critical communication problems. All input (video) and output (motor angles) data are stored to the data server for later analysis.

Note that this PoRoS system is just one example of realizing the Possessed Robot method and we can select other inputs (motion-captured data by the player) and output method (joystick) for other implementations.

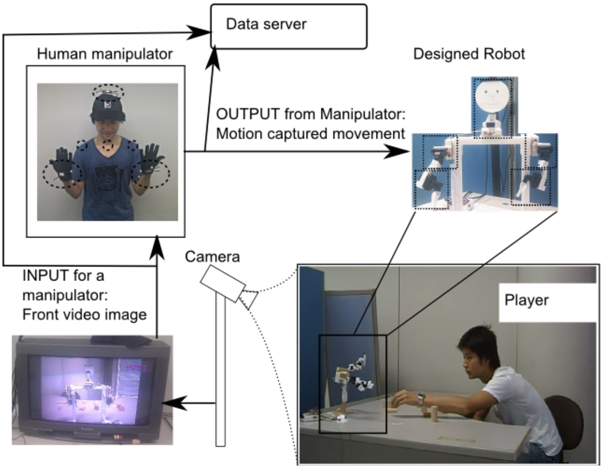


Fig. 5. System implementation

5 Experiment

To research how our method evaluates the design and modalities of a nonhumanlike robot, we compared human-humanlike robot and human-nonhumanlike robot interaction using the PoRoS robot. In nonhumanlike robot interaction, we fixed the head of the robot to decrease the modalities for confirmation. We also prohibited verbal communication during interaction to emphasize the role of the head.

As a demonstrative task, we also setup the assembly of wooden blocks to evaluate our method.

5.1 Pre-evaluation for Creating Evaluation Method

Humans nod for confirmation. Nodding is conducted by the human head. Head nodding has a regulatory role in turn-taking in human-human communication and human-computer interaction [23]; [24].

At first, we examined what kind of procedures humans apply to make buildings by observing human-human interaction. We gathered six participants for this evaluation and assembled three sets of pairs from them. One of the members of a pair took the role of the manipulator. Another member took the role of a player. The manipulator instructed the player to build three kinds of buildings as shown in Fig. 6. All examples in Fig. 6 consisted of five kinds of blocks. First, the manipulator watches the buildings in Fig. 6. Second, she/he sat down in front of the player. Last, she/he instructed the player how to construct the buildings. All manipulators were prohibited to directly touch the blocks. The number of instructions during the evaluation is between five and eight and the construction time is between 30 s and 60 s. The result confirmed that human-human interaction is based on turn-taking strategies. Each pair's turn-taking happened according to each user's nodding and shaking motion.

In detail, the processes are as follows. In the first turn, the player pointed out one of the blocks. If the block was the right one, the manipulator nodded and communication continued to the next turn. If the block was wrong, the manipulator shook her/his head and the player repeated the first turn. In the second turn, the player brought the block to the manipulator and the manipulator directed the player to rotate the block. Then, the player put the block on the building. If the placed position and direction was right, the manipulator nodded and communication returned to the first turn until they completed the building. If the position or direction was wrong, the manipulator shook her/his head. Then, the player placed the block on the desk and repeated the second turn.

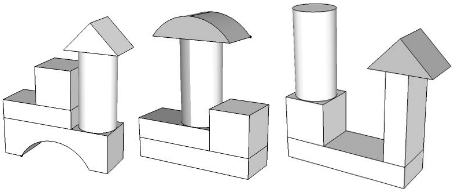


Fig. 6. Example buildings

5.2 Evaluation Method and Hypothesis

Based on the findings from the previous sections, we compared the humanlike group and head-fixed group for validating the proposed method. In the humanlike group, the manipulator could handle the PoRoS robot without any restrictions. However, in the head-fixed group, the neck motor switches were turned off by the system and the manipulator could not control them.

This restriction forced both manipulator and player to use other confirmatory behaviors for turn-taking or it forced both persons to use different communication strategies. When they selected communication strategies other than the turn-taking method, the confirmatory behavior decreased in the head-fixed group.

5.3 Environment for the Experiment

The experimental setup is shown in Fig. 7 left. The manipulator and the player are in separate rooms. The robot is fixed on a desk and placed in front of the player. There are eight blocks on the desk between the player and the robot. The viewpoints of the camera and the robot are located in the same direction. The manipulator can confirm the face of the player. All input and output data are recorded and stored in the data server for later analysis.

We show the scene of manipulation in Fig. 7 right. The manipulator is standing on the left side of Fig. 7 right. Motion-capturing cameras surround him. The video screen is in front of the manipulator and the screen shows the robot, the blocks, and the player as shown in the right top part of Fig. 7 right. An image of the building is pasted on the right side of the screen, and the manipulator instructs the player how to assemble the blocks via the robot.

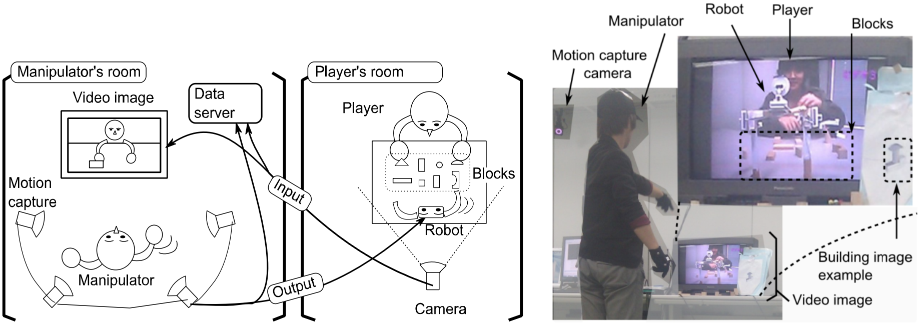


Fig. 7. (left) Experimental setup (right) Experimental scene

5.4 Participant and Experimental Flow

Thirty-six participants participated in the experiment. There were 34 males and 2 females. We assigned 18 participants (including one female) to the humanlike group and the remaining 18 participants to the head-fixed group. Eighteen participants on each group were paired (a manipulator and a player). Each group had nine pairs.

The experiment was divided into the testing phase and the recording phase. Before the experiment, we instructed the participants as follows: "In this experiment, you need to create general communication strategies for the robot with the assembling task. Do not use any kind of code that is incomprehensible to other person." This instruction served the purpose to keep the designed communication strategies general.

At first, each manipulator calibrated the robot parameters to the scale of his/her body. Then, the pairs started the testing phase. During this phase, each manipulator gave instructions for any kind of buildings she/he could imagine. The members in each pair made trial-and-errors efforts and improved their communication strategies.

When the pair determined that they could not improve their manipulation time anymore, the experiment moved to the recording phase. We assigned the manipulator one of the three examples in Fig. 6 and recorded the interaction. The pair required to assemble the building within 300 s. When the recording finished, each participant answered the questionnaire and the experiment was terminated.

5.5 Prediction: Overlapped Time Ratio and Confirmation Ratio

Pre-evaluation confirmed that turn-taking behavior was used in human-human interaction with instructions on how to assemble the blocks. The evaluation also revealed that head movement played a key role on regulating turn taking. However, turn taking itself is difficult to evaluate by video recording data, especially when this evaluation lacks verbal cues.

We used the overlap time ratio as an indicator of turn-taking behavior between each manipulator and player. A previous HRI study using humanoids showed that the increase in overlapped verbal cues of both persons suggests failure of turn taking [25]. We extended this idea to nonverbal situations. If turn taking took place without any problems, the behavior of the robot and the human did not overlap. In contrast, if turn taking did not succeed, the overlapped time ratio increased. In this paper, we defined overlapped time ratio as robot's moving time during user's lifting per user's lifting time. Note that the failure of turn taking does not directly mean failure of communication. If the task is successfully completed, this increased overlapped time suggests different communication strategies between the manipulator and the player.

We used the player's lifting block time to monitor the player's behavioral time. We counted the behavioral time from the input video-recorded data. We used the robot's moving time to monitor the manipulator behavioral time. When the motor moves more than ten angles in 1 s, we counted this as the behavioral time of the manipulator. The behavioral time of the player did not include the suspending time in air. However, if there was a difference in the overlapped time between the humanlike and the head-fixed group, this difference suggested that the two groups used different turn-taking methods.

Our predictions for the head-fixed group in comparison with the humanlike group are the following:

- Prediction 1: The overlapped time ratio will increase depending on the failure of the turn-taking behavior.
- Prediction 2: The ratio of confirmatory behaviors will decrease.

In the head-fixed group, we asked the manipulator questions such as "Did you use confirmatory behavior? If so, what kind of confirmation did you use?".

6 Results

One male pair in the humanlike group and two male pairs in the head-fixed group could not finish assembling the blocks. Other pairs succeeded in this task.

The average overlapped time ratio in the humanlike group is .608 (SD = .062). The average overlapped time ratio in the head-fixed group is .761 (SD = .125). We applied the Welch t-test to both groups and the p-value is .0043 < .05. This statistical result shows that the overlapped time ratio in both groups is significantly different. This result supports the first prediction. The overlapped time ratio is shown in Fig. 8. When we removed the failed pairs, the average overlapped time ratio in the humanlike group is .792 (SD = .132) and the overlapped time ratio in the head-fixed group is .132 (SD = .151). The p-value from the Welch's t-test is .01 < .05, which also suggests significant difference.

The questionnaires after the experiment showed that all manipulators in the humanlike group used head nodding and shaking for confirmation. In contrast, nine manipulators in the head-fixed group raised their hand for confirmation and shook their hand for denying. Two manipulators in the head-fixed group answered that they did not use confirmation in their communication. Based on this result, we counted the raising and shaking hands as confirmation in the head-fixed group.

The players use two kinds of confirmations before and after lifting the blocks. Confirmation before lifting the blocks (before-confirmation) was used to point which block is right or wrong. Confirmation after lifting the blocks (after-confirmation) was used to point which location and direction is right or wrong. We counted both confirmations.

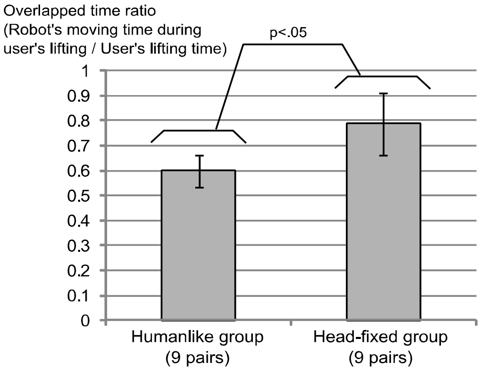


Fig. 8. Overlapped time during humanoid and hand robot

The average before-confirmation ratio is .63 (SD = .22) in the humanlike group and .09 (SD = .19) in the head-fixed group. We applied Welch's t-test to both groups and the results showed p-values of .00003, which is less than .0001. When we removed the failed pairs, the average before-confirmation ratio is .62 (SD = .22) in the humanlike group and .11 (SD = .20) in the head-fixed group. The p-value of the Welch's t-test is .0006, which is less than .001 and suggests significant difference.

The average after-confirmation ratio is .78 (SD = .21) in the humanlike group and .30 (SD = .24) in the head-fixed group. We applied Welch's t-test to both groups and the result showed p-values of .0005, clearly smaller than .001. When we removed the failed pairs, the average before-confirmation ratio is .78 (SD = .23) in the humanlike group and .28 (SD =.25) in the head-fixed group. The p-value of the Welch's t-test is .001 < .005, suggesting significant difference.

We also counted the manipulation time including the before- and after-confirmation of the robot and the lifting time of the player. The average time is 7.7 s (SD = 2.4 s) in the humanlike group and 12.8 s (SD = 5.0 s) in the head-fixed group. We applied Welch's t-test and found significant difference (p = .017 < .05). When we removed the failed pairs, the average time is 7.1 s (SD = 1.8 s) in the humanlike group and 13.3 s (SD = 5.4 s) in the head-fixed group. The p-value of the Welch's t-test is .02 < .05, suggesting significant difference.

In contrast, the average lifting action is 10.9 (SD = 6.0 s) in the humanlike group and 13.2 (SD = 10.8 s) in the head-fixed group. We applied Welch's t-test and found no significant difference (p = .58 > .05). When we removed the failed pairs, the average lifting numbers were 9.1 (SD = 3.0 s) in the humanlike group and 8.4 (SD = 2.9 s) in the head-fixed group. The p-value of the Welch's t-test is .65 > .10, which suggests no significant difference.

7 Discussion

7.1 Predictions

We found significant differences in the overlapped time ratio and confirmation ratio with and without the failed pairs. These results support our predictions.

Pairs in the humanlike group follow the player-first protocol. After the lifting motion, the player sometimes skipped to check the movement of the robot when they rotated a block and placed it. Confirmation by the robot is sent after the placement in this case. The manipulator usually confirmed every movement of the player. In eight pairs of the humanlike group, the manipulator first pointed the target, the player subsequently pointed the same target, and then the robot confirmed. The failed pair skipped first pointing and it caused more misses. They spent their entire 300 s and the task failed. The recorded video also shows that almost player used turn-taking style strategies because the player watched the robot periodically.

In contrast, the pairs in the head-fixed group follow the robot-first protocol. The manipulator in the head-fixed group sometimes omitted the before-confirmation. In this case, when the robot pointed to a block and the player took it, the player moved the block while observing and following the movement of the robot's arms without any confirmation. The manipulator also omitted the after-confirmation and moved on to the next block. However, omission is happened more in before-confirmation than in after-confirmation. The recorded video also supports that they used following the robot strategy because the player carefully watched the robot during the lifting time.

The manipulation time including lifting time significantly increased in the head-fixed group more than the humanlike group. Based on the video recording, this result suggests that each manipulation time increased in the head-fixed group because they watched the robot motion and followed it. The insignificant difference on the lifting action suggests that the assembling order process is not influenced by the change of modalities. These two results suggest that the change in the head modality did not drastically change the entire communication strategy only the manipulation strategy from the turn-taking style to following the robot style.

These findings support our hypothesis that the turn-taking strategy changed in the head-fixed group. In the head-fixed group, they used robot-leading strategies. We estimate that the limited confirmation modalities forced the pairs to use robot-leading interaction.

7.2 Discussion about the Design Process

The entire design process discussed in Section 3 supports the fact that we can have an alternative communication strategy for nonhumanlike robots using the Possessed Robot method.

The Possessed Robot method shows the potential power of the human computation in robot design. The human brain is the most intelligent computer we can access. It has the most flexible learning and most sophisticated communication algorithms. It can provide the most appropriate response to unpredicted situations. For example, we estimated that the manipulator needed a lot of calibration time even for the motion-capturing system. However, the manipulator quickly customized to the robot body and could behave as if she/he was robot.

We also made variations of design process by different usage of human resources. Participants' free-writings in the questionnaire suggests that swapping the manipulator and the player during the design process will reduce the thinking time. The questionnaire from the manipulator also suggests that usage of a third person who does not know the purpose will increase the generality of the strategy.

7.3 Limitations and Future Work

The purpose of this paper is to evaluate the validity of our method by assembling a block task. Our results show one example of the head-fixed design with no verbal cues leading the robot-first instructions. From the experimental conditions, we infer that this change in the communication strategies is caused by the lack of confirmatory modalities in the head-fixed robot. Our experiment only uses nonverbal communication. Our findings may be useful if the field where verbal interaction costs lead to high cognitive load (like rescue and guiding robots). However, the result cannot be directly applied to human-robot interaction studies if verbal cues are used.

Our findings from the experiment may need further research to show their general applicability, however, our method validates the usefulness of the Possessed Robot method in HRI studies because it can find different communication strategies in human-nonhumanlike robot interaction. Such different strategies are hard to find in the previous approaches that designed and implemented robot shapes and modalities according to human-human interaction. Our results suggest that the robot-leading design may be optimal in the case of headless or head-fixed design robots, such as SmartPal and BIRON [7][8]. It is also possible to assemble guidelines (what design is reasonable and what design is unpredictable) using Possessed Robot method. These guidelines reduces useless investment for development of robot's interface.

In future, we also need to discuss how to find optimal ways to connect the robot I/O to human I/O. In this experiment, we started our simplified demonstration from the viewpoint of decreased human design. Even if the human is a powerful problem solver, we estimate that it is still difficult to handle additional input and output that do not come to humans natively. We predict that studies about prosthesis and augmented human technologies will expand the possibility of human scale.

8 Conclusions

We proposed an alternative approach called the Possessed Robot method to find a robot's unique communication strategy. Previous robot shapes and modalities are designed by imitating human-human interaction. This approach has restricted robot design and behavior within the limitations of the possible human modalities. In our approach, the human manipulator behaves as if she/he possesses the robot and finds the optimal communication strategies based on the shape and modalities of the robot.

We implemented the Possessed Robot system (PoRoS) including a reconfigurable body robot, an easier manipulation system, and a recording system to evaluate the validity of our method. We evaluated the block-assembling task by PoRoS with turning on and off the modality of the robot head.

Synchronized motion significantly increased in the head-fixed design, and the ratio of confirmatory behavior significantly decreased. Based on the results, we find an example case for the optimal communication strategy in the head-fixed design. In this case, the robot leads the users and the user follows the robot compared with the turn-taking communication style in the humanoid condition. This result shows the feasibility of the Possessed Robot method in finding the appropriate strategy according to each robot design.

Acknowledgements. This work was supported by the JST PRESTO program.

References

1. del Pobil, A.P., Sudar, S.: Lecture Notes of the Workshop on the Interaction Science Perspective on HRI: Designing Robot Morphology. ACM/IEEE Human Robot Interaction (2010), `http://www.robot.uji.es/research/events/hri2010`
2. Blow, M., Dautenhahn, K., Appleby, A., Nehaniv, C., Lee, D.: Perception of Robot Smiles and Dimensions for Human-Robot Interaction Design. In: Proceedings of the 15th IEEE Int Symposium on Robot and Human Interactive Communication, pp. 469–474. IEEE (2006)
3. Kanda, T., Kamasima, M., Imai, M., Ono, T., Sakamoto, D., Ishiguro, H., Anzai, Y.: A humanoid robot that pretends to listen to route guidance from a human. Autonomous Robots 22(1), 87–100 (2007)
4. Lee, M.K., Kiesler, S., Forlizzi, J., Srinivasa, S., Rybski, P.: Gracefully mitigating breakdowns in robotic services. In: Proceedings of Human Robot Interaction, pp. 203–210 (2010)
5. Shiwa, T., Kanda, T., Imai, M., Ishiguro, H., Hagita, N.: How Quickly Should a Communication Robot Respond? Delaying Strategies and Habituation Effects. International Journal of Social Robotics 1(2), 141–155 (2009)
6. Hirai, K., Hirose, M., Haikawa, Y., Takenaka, T.: The development of honda humanoid robot. In: Proceedings of the IEEE Intl. Conf. on Robotics and Automation (ICRA), pp. 1321–1326 (1998)
7. Li, S., Kleinehagenbrock, M., Fritsch, J., Wrede, B., Sagerer, G.: BIRON, let me show you something: evaluating the interaction with a robot companion. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, vol. 3, pp. 2827–2834 (2004)
8. Matsukuma, K., Handa, H., Yokoyama, K.: Vision-based manipulation system for autonomous mobile robot 'SmartPal'. In: Proceedings of Japan Robot Association Conference, 3D28 (2004)
9. Matsumoto, N., Fujii, H., Goan, M., Okada, M.: Minimal communication design of embodied interface. In: Proceedings of the International Conference on Active Media Technology (AMT), pp. 225–230 (2005)
10. Bartneck, C.: eMuu – An InterFace for the HomeLab. In: Poster at the Philips User Interface Conference, UI 2002 (2002)
11. Yamaji, Y., Miyake, T., Yoshiike, Y., De Silva, P.R.S., Okada, M.: STB: human-dependent sociable trash box. In: Proceedings of Human Robot Interaction, pp. 197–198 (2010)
12. Shibata, T., Mitsui, T., Wada, K., Tanie, K.: Subjective Evaluation of Seal Robot: Paro – Tabulation and Analysis of Questionnaire Results. Journal of Robotics and Mechatronics 14(1), 13–19 (2002)
13. Fujita, M., Kitano, H.: Development of an Autonomous Quadruped Robot for Robot Entertainment. Autonomous Robots 5, 7–18 (1998)
14. Osawa, H., Ohmura, R., Imai, M.: Using Attachable Humanoid Parts for Realizing Imaginary Intention and Body Image. International Journal of Social Robotics 1(1), 109–123 (2009)
15. Osawa, H., Orszulak, J., Godfrey, K.M., Coughlin, J.: Maintaining Learning Motivation of Older People by Combining Household Appliance with a Communication Robot. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5310–5316 (2010)

16. Kelley, J.F.: An iterative design methodology for user-friendly natural language office information applications. ACM Transactions on Office Information Systems 2(1), 26–41 (1984)
17. Steinfeld, A., Jenkins, O.C., Scassellati, B.: The oz of wizard: simulating the human for interaction research. In: Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction (HRI 2009), pp. 101–108. ACM (2009)
18. Kuzuoka, H., Oyama, S., Yamazaki, K., Suzuki, K., Mitsuishi, M.: GestureMan: A Mobile Robot that Embodies a Remote Instructor's Actions. In: Proceedings of Computer Supported Cooperative Work, pp. 155–162 (2000)
19. Glas, D.F., Kanda, T., Ishiguro, H., Hagita, N.: Simultaneous Teleoperation of Multiple Social Robots. In: Proceedings of Human-Robot Interaction, pp. 311–318 (2008)
20. Lee, J.K., Stiehl, W.D., Toscano, R.L., Breazeal, C.: Semi-Autonomous Robot Avatar as a Medium for Family Communication and Education. In: Proceedings of Advanced Robotics, pp. 1925–1949 (2009)
21. Disney, Turtle Talk with Crush (2004), http://disneyland.disney.go.com/disneys-california-adventure/turtle-talk-with-crush/?name=TurtleTalkEntertainmentPage
22. NaturalPoint, OptiTrack s250e (2010), http://www.naturalpoint.com/optitrack/products/s250e
23. Sacks, H., Schegloff, E.A., Jefferson, G.: A simplest systematics for the organization of turn-taking for conversation. Language 50, 696–735 (1974)
24. Cassell, J., Thórisson, K.R.: The Power of a Nod and a Glance: Envelope vs. Emotional Feedback in Animated Conversational Agents. Applied Artificial Intelligence 13, 519–538 (1999)
25. Chao, C., Thomaz, A.L.: Turn Taking for Human-Robot Interaction. In: Proceedings of AAAI Fall Symposium (Applied Artificial Intelligence) (2010)

Evaluating Telepresence Robots in the Field

Amedeo Cesta, Gabriella Cortellessa, Andrea Orlandini, and Lorenza Tiberio

CNR, Italian National Research Council, ISTC, Rome, Italy
{name.surname}@istc.cnr.it

Abstract. Most robotic systems are usually used and evaluated in laboratory setting for a limited period of time. The limitation of lab evaluation is that it does not take into account the different challenges imposed by the fielding of robotic solutions into real contexts. Our current work evaluates a robotic telepresence platform to be used with elderly people. This paper describes our effort toward a *comprehensive*, *ecological* and *longitudinal* evaluation of such robots. Specifically, the paper highlights open points related to the transition from laboratory to real world settings. It first discusses some results from a short term evaluation performed in Italy, obtained by interviewing 44 healthcare workers as possible clients (people connecting to the robot) and 10 older adults as possible end users (people receiving visits through the robot). It then describes a complete evaluation plan designed for a long term assessment also dwelling on the initial application of such methodology to test sites, finally it introduces some technical features that could enable a more robust real world deployment.

1 Introduction

The area of social robotics is receiving increasing attention and the task of "robot as companions" has received attention at research level [1]. Several projects have also proposed different types of solutions with robots that both interact with humans and are connected to heterogeneous technology to build innovative living environments (e.g., [2,3,4]). This paper aims at underscoring one aspect connected to such a line of innovation that deserves special attention: the study of *attitude and perceptions of people who share the environments in which the robot operates* over *long periods of time*.

It is worth noting how in robotics there is a deep-rooted tradition in developing technology usually shown in sporadic events and for short periods, i.e., for demos or live show cases, which are intended to demonstrate the "enhanced" characteristics of a prototype, making them attractive while "hiding" or at least "containing" the technical problems connected with any long term use within a comprehensive application. Indeed, a key requirement for social companions (e.g., robots assisting old people at home) is their continuous operation, their robustness and the continuous interaction with humans over time. Such continuity of use has significant implications on the technology development but it also highlights the need to design a methodology for assessing human reactions with respect to prolonged use of the proposed solutions. The challenges for the Intelligent Technology and the Human Robot Interaction researchers are numerous and mainly related to two aspects: (a) in terms of *users perspective*, robots must adhere to user requirements and be acceptable in the long term, (b) in terms of *technology*, the

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 433–448, 2013.

need exists to create robust, efficient and secure solutions. More specifically, the transition from a use in the laboratory to an actual deployment into real contexts, highlights the need for a transition from short term to long term experiments hence requiring a shift of attention on the aspects highlighted in this paper. In particular we underscore how *long-term use* and *evaluation* are key points to be addressed to ensure that robotic technology can make a leap forward and be used in real environments.

In the framework of the EU Ambient Assisted Living (AAL) Joint Program[1] we are performing a wide program of evaluation in the field of an industrial mobile telepresence platform called GIRAFF produced by GIRAFF Technologies AB[2], Sweden. More specifically, we take part in an evaluation spanning three different EU countries – Italy, Spain and Sweden. The evaluation takes social and psychological factors into account to study users attitude and reaction, but also analyzes the emergence of "undesired behaviors" like technological weaknesses in continuous operation, possibly leading to human rejection. In this work, we present the results gathered in Italy after the short term evaluation phase and, then, we present and discuss the general long term evaluation methodology. The paper introduces the context of work (Section 2), then analyzes and reasons about the work both to realize experiments with real users outside the laboratories and to develop a methodology for addressing long term evaluation (Section 3); finally it describes some technological challenges for telepresence robots when fielded in real environments (Section 4).

2 Context of Work

Telepresence robots have been increasingly proposed to be used in workplace and Mobile Remote Presence (MRP) systems have been studied as a means to enable remote collaboration among co-workers [5,6]. Furthermore, MRP systems are also being used to provide support to elderly people. In this respect, some initial research exists which aims to understand the acceptance of older adults, their concerns and attitude toward the adoption of MRP systems [7,8,9]. Our work is motivated by the participation to an AAL project, called EXCITE[3], aiming at promoting the use of MRP robots to foster interaction and social participation of older people as well as to provide an easy means to possible caregivers to visit and interact with their assisted persons while in their living environment. GIRAFF (see Figure 1) is a remotely controlled mobile, human-height physical avatar integrated with a videoconferencing system (including a camera, display, speaker and microphone). It is powered by motors that can propel and turn the device in any direction. An LCD panel is incorporated into the head unit. The robotic platform is accessed and controlled via a standard computer/laptop using a software application. From a remote location the *client* (member of family or healthcare professionals) with limited prior computer training teleoperates the robotic platform while older people (*end users*) living in their own home (where the robot is placed) can receive their visit through the MRP. The remote user can charge the robot batteries by driving it onto a docking station.

[1] `http://www.aal-europe.eu/`

[2] `http://www.giraff.org`

[3] `http://www.excite-project.eu/`

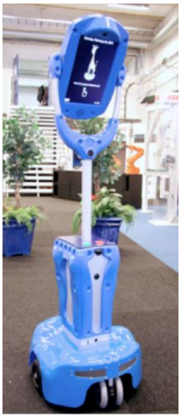


Fig. 1. The GIRAFF robot

Key Pursued Ideas. The ExCITE project aims at assessing the validity of a MRP system in the field of elderly support in different European countries. The project innovative concepts are the following:

- *User Centered Product Refinement*. This approach is based on the idea of obtaining users feedback during the time they use the robot and cyclically refine the prototype in order to address specific needs;
- *User tests outside labs*, rather then testing the system in laboratory setting, the robotic platform is placed in a real context of use. This approach is in line with several research that highlights how systems that work well in the lab are often less successful in real world settings [10]. The evaluation of robots made in a laboratory environment does not favor the emergence of robotic aid suitability to support elders who are able to stay in their own homes. For this reason an essential step is to assess the technology in the specific contexts in which the technology is supposed to be used [11];
- *Use on a time period long enough*, to allow habituation and possible rejection to appear. Indeed, interviews and survey conducted after a short period of time can be limited and can prevent other effects to emerge. On the contrary, a key aspect of relationship is that it is a persistent construct, spanning multiple interactions [12]. In this light, in order to assess the human-robot interaction it is important to investigate how people interact with robots over long periods of time.
- *Analysis of cultural and societal differences*, an interesting part of our project stems in the idea of comparing the long term deployment of the telepresence platform in different countries so as to allow an analysis of cultural and societal differences over European countries.

Different GIRAFF prototypes are being deployed for long periods of time (at least three months, and possibly 1 year) in real context of use. Feedback obtained from the users (both older users having the robot at home and the clients, that is people connecting and visiting the older) is used to improve the robot. In what follows, we describe our progressive work toward a long-term human-robot interaction assessment and then we discuss some lessons learned from a more focused technological point of view.

3 The Evaluation Effort

We have conceived a twofold path for evaluating the human-robot interaction gathering both feedback from short interactions between potential users and the GIRAFF robot and also focusing on a long term assessment plan. More specifically we identified two tracks for our effort:

– *Short Term Evaluation*, which consists of a collection of immediate users feedback (i.e., after a short interaction with the robot) on the telepresence robot, connected to different aspects of the interaction mainly related to the usability, willingness to adopt it, possible domains of applications, advantages and disadvantages.
– *Long Term Evaluation*, which relates to the study of the long-term impact of GIRAFF's social and physical presence on elderly users using the system both to communicate with their relatives and friends and to receive visits from healthcare providers and in general caregivers.

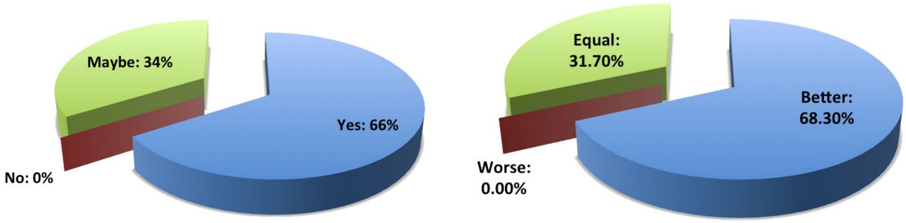
The short term evaluation effort, though not sufficient alone, still provides immediate feedback that can be used to quickly improve the technological development, to possibly add functionalities to the system or to simply confirm the validity of some technological choices. In addition it can give valuable guidance to the long-term assessment. For this reason we adopted a combined approach involving participants representative of both types of users: the *client side* and the *end user side*.

Following this schema, we first present results for the short term evaluation performed in Italy, then we introduce our complete design for a methodology to assess the long-term impact also reporting the status of the Italian long-term test sites that are currently running according to this methodology.

3.1 Short Term Evaluation

As stated in [7], before intelligent technologies would be accepted, it is important to understand their perception of the benefits, concern and adoption criteria. In our study, we aim at reproducing as much as possible an "ecological" setting for the experiment. To this purpose we distinguished the role of the users and recruited different participants according to their expected role. Specifically for the client side, we recruited users representative of the potential visitors of the elderly users among caregivers, nurses, health workers, etc. For the end user side we interviewed older adults living alone, or possibly receiving some kind of health care assistance. In the following we describe the general method and the material and procedure used for both types of users.

Method. This evaluation was aimed at assessing users reaction toward the possible adoption of the GIRAFF system as a means to visit or provide some kind of service to the elderly users. Aspects investigated were *willingness to adopt the robotic solution*, *possible domains of application*, *advantages* and *disadvantages* and *suggestions for improvements*.



(a) Willingness to adopt GIRAFF

(b) GIRAFF vs other teleconference systems

Fig. 2. General assessment of the GIRAFF system

Health Workers as Clients

Participants and Procedure. Forty-four health workers[4] from different specialist areas were recruited for this study. The sample interviewed so far is composed by 26 women and 18 men with a mean age of 42 years, $SD = 12.2$.

The meeting entailed a tutorial presentation of 20 minutes to describe features and functionalities of the telepresence robot. After this tutorial, a practical session allowed the health workers to operate the system and experience the different functionalities. Following the tutorial a focus group was conducted and a final questionnaire was administrated to assess possible applications of the telepresence robot, the perceived advantages and disadvantages of the system, the patient profile best suited to benefit from the use of an aid-based on telepresence.

Results

General Assessment. A first analysis of the results showed a positive reaction of the participants to the system. In particular 66% of participants would be willing to use GIRAFF as an aid support in his/her profession as well as no one opposes the use of robots (see Figure 2a). In addition most of them judge the telepresence robot as a better tool with respect to traditional teleconference system like Skype (see Figure 2b).

Profile of Potential Users. Results also identify the categories of people who could benefit from the use of telepresence robots: specifically, the category "self-sufficient or semi-autonomous elderly living alone" has been mentioned by 35% of respondents; 25% of the subjects also indicates "adults and elderly patients in home care and with special needs", such as patients in isolation for infection, dialysis patients or with chronic diseases such as Chronic Obstructive Pulmonary Disease (COPD) or diabetes). A 20% of the responses were grouped into the category "older people with early or mild dementia". Two other categories were "adults or elderly with physical disabilities" (17%) and "young people and adults with intellectual disabilities" (7%).

[4] A preliminary version of this paper [13] reported results with a limited number of subjects. This paper reports on a specifically recruited enlarged sample.

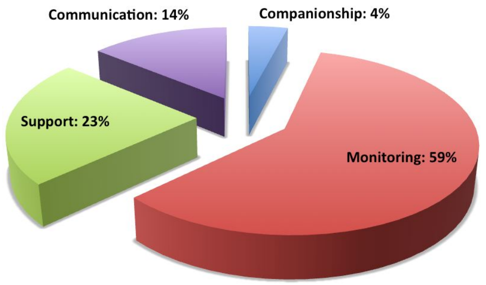


Fig. 3. Favorite GIRAFF's domains of application

Application Domains. The participants are in favor of the use of robots to train the family caregiver to small nursing tasks and to maintain constant contact with assisted people. The possibility of *continuous monitoring* (see Figure 3) of the patient at home is considered the most useful application (59% of participants were in favor of this kind of application), with *support* application following at 23%, while the *companionship* and *communication* functionalities are less envisaged, maybe also due to the specific type of professional activities of participants.

More specifically, 45.5% of the health workers advocate the use of the robot to train a family caregiver to perform small nursing tasks (e.g., treat a bedsore, administer an enema, measuring of vital signs) and to maintain a constant contact with the patient and his family (75% of participants). Finally 60% of participants also says that the robot could alleviate the workload of the family caregiver, but not that of the health workers themselves (50% of people admit to be uncertain about the real possibility of the robot to diminish their daily workload).

Advantages and Disadvantages. Among the advantages, participants listed the ability to monitor remotely *via* visual communication the physical state of health, as well as the possibility to follow the management of medication and certain health practices (control of vital parameters such as level of blood glucose for diabetic patients, supervision of practices related to their care and medication as deep breathing exercises for patients with COPD). One advantage often cited by the health workers is the possibility for the operator to improve his/her night surveillance activity in hospital and home care cases.

Suggested Improvements. The focus group conducted at the end of this analysis, highlighted some aspects considered as particularly relevant for using the platform in the healthcare domain for long-term period. These aspects specifically refer to improvements and integration of additional functionalities. Specifically according to participants, there is a need to improve the video quality, especially in relation to night vision; it would be useful to add the zoom functionality to the webcam; the duration and mode of charging should be improved: e.g., it would be better that the robot could reach autonomously the re-charging station; the safe navigation of the robot should be guaranteed (see Section 4 for a more detailed discussion). In addition it would be beneficial to enable the call transfer if the client is not connected to the robot via the PC; finally the transmission of vital parameters to the doctor should be supported. All these suggestions for technical improvements are currently inspiring the future modifications of the GIRAFF system in line with the user centered approach pursued in the project.

Older Adults as End-Users

Participants and Procedure. To investigate aspects connected to the end-user interaction with the telepresence system we contacted 10 older adults. Four of them were potential end users who have been asked to participate in the long-term evaluation described later in this paper. The remaining participants are involved in a parallel study, also connected to the project that aims to validate the GIRAFF system as a tool for providing remote rehabilitation [9].

The procedure followed in this qualitative research entailed an explanation of the main idea underlying the telepresence system, showing some descriptive materials, a video of the system and, where possible, a practical demonstration of the system itself. The selection of the material and the modality to present the system were decided according to the time availability, and the specific situation presented in each evaluation session. Overall, we here opted for a qualitative analysis given the relatively small number of the sample.

Results. A qualitative analysis of the interview have been conducted and the most relevant feedbacks are here reported in terms of positive and negative aspects of the robotic system.

Positive Aspects. Among the positive aspects most of the subjects reported the following: participants judged the visit through GIRAFF as engaging and “real”; the robot was pleasant to see; the ability of the robot to move in the environment was positively assessed; users felt physically involved during the interaction; participants think that the robot would help someone living alone at home to feel safer; participants judged positively both the audio and the video functionalities; participants think that interaction through the robot was spontaneous.

Negative Aspects. Among the most negative aspects we mention: the GIRAFF system is too big and consequently may not be well integrated in a domestic environment due to its size; the battery power may be too short; there may be some problems due to the privacy issue; there were some concerns related to the safe movement of the robot and to its ability of obstacle avoidance; some “intelligent features”, like the autonomous recharging of the battery, are missing; the connection to the docking station is not very intuitive.

Also this effort showed an overall positive reaction to the system, even though some improvements are desired in view of a real usage of the system. It is worth underscoring how the key point here is the fact that qualitative data has been gathered by interviewing “real potential users” like for example a group of caregivers and old people that can receive visits through the robot.

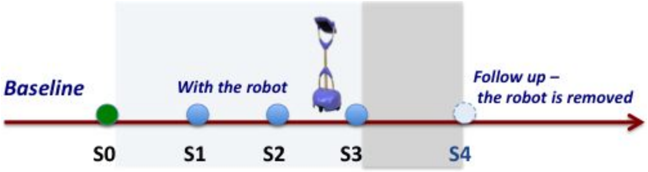


Fig. 4. The Long Term Evaluation timeline

3.2 Long Term Evaluation

One of the original features of the ExCITE project consists of realizing long-term experiments involving older people hosting the robot in their living environment both to communicate with others and to receive assistance services.

Method. Figure 4 gives a general idea of the designed method to evaluate features over time. The evaluation entails a period of N months (with $3 \leq N \leq 12$) during which the end user will have the robot at home and the clients can visit him/her through it. Assessment happens at milestones S_i. Specifically, after an initial assessment (S0 in figure) at the beginning of the experimentation (*baseline*), the variables of interest are measured at regular intervals (S1-3) to observe changes over time. At the last month the GIRAFF will be removed from the end user apartment and the same variables will be assessed again after 2 months from this removal (S4). The general idea is to use a repeated measures method to see changes over time during the long term usage of the robot.

Participants and Procedure. Three different cases have been identified to cover different situations in which the robot can be deployed. Specifically, for the client typology of users we considered (a) a *formal caregiver* belonging to an Health care organization; (b) a *family member (caregiver)*; (c) *other relatives or friends* who may visit the elderly person through the robot. The type of material used in the long term evaluation for both the client and the end user depends upon the type of interaction for which the telepresence is used. For this reason, for each of the three mentioned situations we had developed (or selected) a set of questionnaires (almost all validated in the three languages of the involved countries) aimed at monitoring specific variables and to be administrated at specific time both to end users and to clients.

Material. For each of the described case we prepared the material to assess the variables under study at the specified intervals. Table 1 lists in detail the different variables and the related instruments to be used to measure the variables over time.

Client Side. Specifically on the client side, during the initial step (S0), we use: (a) an informed *consent form* describing the aim and procedure of the study; (b) the *socio demographic data* form to gather some relevant information on the user; (c) we developed on purpose a questionnaire aimed at assessing the client expectation on the GIRAFF's ability to ease the support (*Support Expectation*). It is worth highlighting that we developed two slightly different types of questionnaires for the formal and informal caregivers, while for the third type of client we designed a questionnaire (*Influence on Relationship Expectation*) on the expectation on GIRAFF as a means to ease and support the remote communication and consequently the social relationship.

During the following step (S1), for all three types of clients we will use: (a) questionnaires based on the SUMI inventory [14] to assess the *usability* of the client software: (b) will ask participant to keep a *diary* to register the "salient" events of the visit through telepresence in terms of encountered problems, good features and so on.

During the subsequent step (S2), in addition to the diary that clients have to keep along the whole experience with the robot, we make a first assessment of ability of GIRAFF to ease the support (or the communication) between the client and the end user through the *Support Assessment* and *Impact on Relationship Assessment* questionnaires. In addition, during this phase we will also use the Temple Presence Inventory [15] that is a tool to measure dimensions of (tele)presence.

At step S3 we use the Positive Affect Negative Affect Scale, PANAS, [16], and a final structured interview to assess the overall experience in terms of the most relevant variables considered in the study.

After two months from the robot removal, S4 will allow assessing the impact of its absence through the *Support Assessment* questionnaire.

End User Side. For the end user receiving the robot we followed a similar approach, but we focused on some additional variables that is worth dwelling on. Specifically, we measure (a) the *perceived loneliness* through the UCLA Loneliness Scale [17], which was developed to assess subjective feelings of loneliness or social isolation;(b) the perceived health status through the Short Form Health Survey (SF12) [18]; (c) the Multidimensional Scale of Perceived Social Support [19]; (d) Geriatric Depression Scale [20]: a modified version of the Health Service Satisfaction Inventory. Finally the Almere [21] model will allow assessing dimensions of technology acceptance.

In the table, measures highlighted in bold will ensure the repeated measures thus allowing to observe the influence of changes over time. It is worth underscoring how this evaluation methodology will allow monitoring the human-robot interaction over time, thus contributing to understand the long term impact of a fully deployed robotic solution.

The actual implementation of this plan in three different European countries will also support a cross-cultural analysis, continuing some work started on this specific topics [22]. The following subsection briefly reports on the current status of the Italian test sites.

3.3 First Test Sites Running

Two test sites have started in Italy that are representative of the *family-member-elderly* user category.

Test Sites 1. A couple of old people living in the countryside near Rome are the end users of this test site (see Figure 5). The man has reduced mobility, while the woman has problems with her sight. They are quite independent although their health condition

Table 1. Long term evaluation: variables measured along the phases (S0–S4) and related material

PHASES	**S0**	**S1**	**S2**	**S3**	**S4**
Client					
Health Professional	Consent Form, Socio-Demographics Data Form, **Support Expectation**, **Diary**	Usability, **Diary**	**Support assessment**, Temple Presence Inventory, **Diary**,	PANAS, PIADS, Final Interview, **Diary**	**Support Assessment**
Family member	Consent Form, Socio-Demographics Data Form, **Support Expectation** (informal carer), **Diary**	Usability, **Diary**	**Support assessment** (informal carer), Temple Presence Inventory, **Diary**	PANAS, PIADS, Final Interview, **Diary**,	**Support Assessment** (informal carer)
Relatives friends	Consent Form, Socio-Demographics Data Form, **Influence on Relationship** **Expectation**, Diary	Usability, **Diary**	**Influence on Relationship assessment** (informal carer), Temple Presence Inventory, **Diary**	PANAS, PIADS, Final Interview, **Diary**	**Influence on Relationship Assessment**
End User					
Elderly	Consent Form, Socio-Demographics Data Form, **Loneliness (UCLA)**, **Short Form Health Survey (SF12)**, **Multidimensional Scale of Perceived Social Support**, **Geriatric Depression Scale**, **Almere model**, **Health Service Satisfaction Inventory** (if applies)	**Loneliness (UCLA)**, **Multidimensional Scale of Perceived Social Support**, **Geriatric Depression Scale**, Attitude_Acceptance, **Health Service Satisfaction Inventory** (if applies)	Temple Presence Inventory, **Almere model**	**Loneliness (UCLA)**, **Short Form Health Survey (SF12)**, **Multidimensional Scale of Perceived Social Support**, **Geriatric Depression Scale**, **Almere model**, PANAS, PIADS, Final Interview	**Loneliness (UCLA)**, **Short Form Health Survey (SF12)**, **Multidimensional Scale of Perceived Social Support**, **Geriatric Depression Scale**, **Health Service Satisfaction Inventory** (if applies)

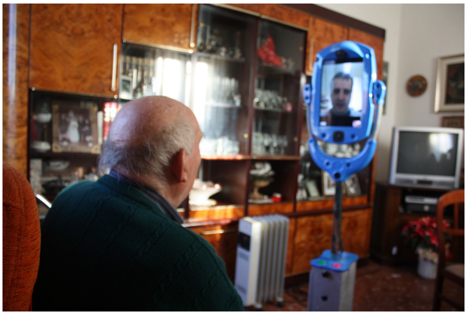


Fig. 5. A picture from the first Italian test site

is slowly deteriorating. The secondary users are: their son living in Rome and their grandchild.

We initially experienced some problems with the technical set-up of this test site. Specifically, the typical layout of the Italian houses has created some problems due to reduced space (particular difficulty emerged in going through doors and due to some narrow passage in the house) to the connection to recharging station and to smoothly move in the house. This highlighted the need to improve the mobility of the robot and the need to provide an automated recharging functionality (see Section 4). Currently the test site is at step S0 of the evaluation plan. Since now we can say that some robot usability problems are emerging due to the particular fragility of the two old people who participate in the study. Overall the old couple are very interested to the GIRAFF robot, even though its use is currently still limited. Our goal is also to monitor the robot's usage over time to assess the effect of familiarity or habituation.

Test Sites 2. A very active old woman living alone in Rome is the end user of our second Italian test sites. Her grandchild and daughter are the main current secondary users. Additionally we are also planning to involve a day care center that will connect to the woman. Also this test site is currently at step S0 of the evaluation plan. However, some preliminary comments can be reported. Both the lady and her grandchild are enthusiastic of the robot. They would also like that the robot do additional things. The lady, as most of the elderly people interviewed, is concerned about possible costs associated to the robots (e.g., the electricity consumption). Overall she really appreciates the possibility to stay in contact with her relatives, also relying on the video capability of the robot. She would also appreciate a sort of service provided by the day care center that would allow her to have a more frequent contact with a doctor or a specialist.

4 Technical Challenges for Long Term Use

The work performed for both the short term experiments and the test sites set up reinforced the general lesson that in order to evaluate MRP systems, an important aspect worth to be considered is the effort needed to deploy them into real contexts. In particular, the technological set up needed to deploy the systems in real older people houses requires quite an amount of work to be done in order to create robust contexts of use.

In this regard, even at an intermediate stage of the ExCITE project, some general comments can be derived from such an experience.

Specifically, in our work of fielding the telepresence robot in operational contexts in Italy, we gathered incremental evidence that situations exist in which some technical advancements usually connected with autonomous behavior can dramatically affect the efforts required in deploying the robot in older people living environment. And, most importantly, they also enhance the interactions through the MRP robot as well as increase the robustness of the whole system in an application area where frail users are involved. Indeed, such advancements can be related to human-robot interaction factors [23]: (a) the *system effectiveness*, the quality of interactions related with the robot capability of performing some tasks autonomously; (b) the *system efficiency*, the amount of time spent in order to complete some instrumental robot activities; (c) the *operator workload*, the cognitive burden required to the operator while controlling the MRP robot; (d) the *robot self-awareness*, the capacity of the robot to assess its own status; (e) the *robot human-awareness*, the degree of which the robot is aware of human presence in the environment. Then, from the client users' view, even a limited set of autonomous behaviors can increase their projection capability and achieve a safe and reliable operation of the telepresence robot in a (potentially) dynamic environment [24].

4.1 Gaining Robustness by Introducing Autonomous Capabilities

In this subsection, we present some contextualized use cases for autonomous capabilities that have emerged from our experience.

Autonomous Navigation. A quite common situation in Mediterranean countries is related to the small size of apartments where old people live. In fact, the GIRAFF robot is currenlty being installed and operated also in domestic places of very limited size. This entails that even skilled client users may have difficulties in controlling the robot within such small environments. Therefore, the system should be equipped with navigation abilities to safely guide the telepresence robot to some specific home locations requested by the operator. For instance, the operator may request the MRP to reach the kitchen in the apartment so that she can visually check the status of the stoves. In this respect, the autonomous navigation capabilities would clearly enhance both system effectiveness and efficiency thus lowering the operator workload during the overall interaction.

Docking. While deploying the robot at home, a crucial location emerged to be the position of the *docking station* used to recharge the batteries of the robot and park it while idle. In fact, since the robot must not be left out of the docking station without control, this is the most important location that the robot should always be able to reach. The robot should also be aware of the status of its battery (increased robot self-awareness) and, whenever the level is below a given threshold, it should automatically reach the docking station. Indeed, the return to the docking station is considered the aspect that requires synthesis of new solutions. In this case, an autonomous docking capability would surely lower the time spent in performing an instrumental activity.

Connectivity. Another important issue of a long term domestic test site is that very ofter it is not possible to rely on a continuous active WiFi internet connection (in particular, this issue has been detected as a quite critical one in Italy but also in Spain where our partners are running analogous test sites). Some time, sudden communication breakdowns can leave the robot with no active control. This can be risky since the robot may remain stuck in an hazardous position possibly preventing movements of the old person at home. In such cases, the robot should automatically reach and, then, plug into the docking station. Therefore the idea is to minimize the risk of leaving the robot with no charge in the middle of an apartment thus guaranteeing an important enhancement of the system.

People Searching. During an emergency call[5] a client user should take control of the robot and find the elderly as soon as possible in order to check her health conditions. In such cases, the telepresence platform could be endowed with the capability of autonomously looking for the elder in the apartment (i.e., increased robot human-awareness) instead of requiring the operator to (potentially) visit the whole house at random. Then, once the operator takes the control of the robot, it will automatically find the proper position to start a new dialogue, minimizing the movement of the robot in the environment and increasing the effectiveness of the interaction.

People following. During a dialogue, a client user should be allowed to focus her attention to the old person movements/gestures rather than continuously adjusting the robot position. Then, an interesting feature would be to enable GIRAFF to automatically identify the position of the old person (increased robot human-awareness) and/or autonomously adapt its position maintaining the person centered on the robot's camera. This clearly would increase the system effectiveness as well as soften the operator burden, then, enhancing the overall quality of the interaction.

Safety. A final point worth being mentioned is the paramount importance of the safety of platform basic movements to avoid any scaring movement in proximity of the old person (increased robot human-awareness). In this respect recent technology improvements for fault-free low level behavior like those described in [25] could potentially result very useful if integrated.

As a final remark, it is worth underscoring that all the above mentioned cases can be addressed with a smart integration of functionalities that are within the state-of-the-art of current autonomous robotics and, then, such advancements are more a matter of implementation efforts on the MRP platform rather than actual technological development.

5 Conclusions

This paper describes the ongoing work that is trying to assess a robotic telepresence system within the elderly domain. Two important aspects are presented that can be

[5] This is a specific capability that allows to force the usual operational functionalities. It enables an authorized client to bypass the old person authorization and connect with the home environment.

considered as mandatory steps for both a general roadmap in robotics and our specific work.

As a first contribution, we have highlighted the importance of performing *ecological experiments*, i.e., which reproduce as much as possible the actual conditions of use of robotic technology, in terms for instance of real people who use it and real context of use. Although still simple in the results, analysis of the short-term evaluation provides a number of indications "from the field" that are representative of the actual users' expectations, both in relation to the human-robot interaction and to the most urgent technological improvements essential for an effective deployment. For example, health workers expressed a number of requests that would be important to fruitfully use the GIRAFF system as a means to support their work. At the same time, the technological tests done in real homes, highlighted technological barriers that must be necessarily overcome.

The article's second contribution concerns our effort toward a long-term assessment. Other works in the area have highlighted this need but in this article we have proposed a rather elaborated and detailed methodology for the long-term evaluation that is currently being applied to real test sites of elderly people for long periods of time.

In addressing these two points, there were also a number of technological challenges and requirements for "intelligent features" that the technology should incorporate and that could contribute to solve some of the open challenges in moving from a short-term demonstration to a real and continuous use.

Acknowledgements. Authors are partially supported by the EU under the Ambient Assisted Living Joint Program – EXCITE Project (AAL-2009-2-125) – and under the ICT Call 7 – *GiraffPlus* Project (GA 288173). Authors are indebted to project partners for the stimulating work environment. Interactions with the colleagues from Örebro University have been fruitful to synthesize the evaluation plan. Authors would like to thank Vittoria Giuliani for comments to preliminary versions of this work.

References

1. Tapus, A., Matarić, M.J., Scassellati, B.: The Grand Challenges in Socially Assistive Robotics. IEEE Robotics and Automation Magazine 14, 35–42 (2007)
2. Pineau, J., Montemerlo, M., Pollack, M., Roy, N., Thrun, S.: Towards Robotic Assistants in Nursing Homes: Challenges and Results. Robotics and Autonomous Systems 42, 271–281 (2003)
3. Cesta, A., Cortellessa, G., Rasconi, R., Pecora, F., Scopelliti, M., Tiberio, L.: Monitoring elderly people with the RoboCare Domestic Environment: Interaction synthesis and user evaluation. Computational Intelligence 27, 60–82 (2011)
4. Saffiotti, A.: The Concept of Peis-Ecology: Integrating Robots in Smart Environments. Acta Futura 3, 35–42 (2009)
5. Lee, M.K., Takayama, L.: Now, I Have a Body: Uses and Social Norms for Mobile Remote Presence in the Workplace. In: Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems, CHI 2011, pp. 33–42. ACM, New York (2011)

6. Tsui, K.M., Desai, M., Yanco, H.A., Uhlik, C.: Exploring Use Cases for Telepresence Robots. In: Proceedings of the 6th Int. Conf. on Human-Robot Interaction, HRI 2011, pp. 11–18. ACM, New York (2011)
7. Beer, J.B., Takayama, L.: Mobile Remote Presence for Older Adults: Acceptance, Benefits, and Concerns. In: Proceedings of Human Robot Interaction, HRI 2011, Lausanne, CH, pp. 19–26 (2011)
8. Kristoffersson, A., Coradeschi, S., Loutfi, A., Severinson Eklundh, K.: Towards Evaluation of Social Robotic Telepresence based on Measures of Social and Spatial Presence. In: Proceedings on HRI 2011 Workshop on Social Robotic Telepresence, Lausanne, pp. 43–49 (March 2011)
9. Tiberio, L., Padua, L., Pellegrino, A., Aprile, I., Cortellessa, G., Cesta, A.: Assessing the Tolerance of a Telepresence Robot in Users with Mild Cognitive Impairment – A Protocol for Studying Users' Physiological Response. In: Proceedings on HRI 2011 Workshop on Social Robotic Telepresence, Lausanne, pp. 23–28 (March 2011)
10. Sabanovic, S., Michalowski, M., Simmons, R.: Robots in the Wild: Observing Human-Robot Social Interaction Outside the Lab. In: Proceedings of the International Workshop on Advanced Motion Control, Istanbul, Turkey. ACM (2006)
11. Hutchins, E.: Cognition in the Wild. MIT Press (1995)
12. Bickmore, T.W., Picard, R.W.: Establishing and Maintaining Long-Term Human-Computer Relationships. ACM Transactions on Computer Human Interaction 12, 293–327 (2005)
13. Cesta, A., Cortellessa, G., Orlandini, A., Tiberio, L.: Addressing the Long-term Evaluation of a Telepresence Robot for the Elderly. In: Proc. of the 4th Int. Conf. on Agents and Artificial Intelligence, ICAART 2012. Artificial Intelligence, vol. 1, pp. 652–663. SciTePress (2012)
14. Sumi: Software Usability Measurement Inventory, University College Cork (2011), `http://sumi.ucc.ie/`. (last checked November 2011)
15. Lombard, M., Ditton, T., Weinstein, L.: Measuring Telepresence: The Temple Presence Inventory. In: Proceedings of the Twelfth International Workshop on Presence, Los Angeles, California (USA), San Francisco (2009)
16. Terracciano, A., McCrae, R.R., Costa, P.T.: Factorial and Construct Validity of the Italian Positive and Negative Affect Schedule (PANAS). European Journal of Psychological Assessment Official Organ of the European Association of Psychological Assessment 19, 131–141 (2003)
17. Russell, D., Peplau, L.A., Cutrona, C.E.: The Revised UCLA Loneliness Scale: Concurrent and Discriminant Validity Evidence. Journal of Personality and Social Psychology 39, 472–480 (1980)
18. Ware, J.E.J., Kosinski, M., Keller, S.D.: A 12-Item Short-Form Health Survey: Construction of Scales and Preliminary Tests of Reliability and Validity. Medical Care 34 (1996)
19. Zimet, G.D., Dahlem, N.W., Zimet, S.G., Farley, G.K.: The Multidimensional Scale of Perceived Social Support. Journal of Personality Assessment 52, 30–41 (1988)
20. Yesavage, J.A., Brink, T.L., Rose, T.L., Lum, O., Huang, V., Adey, M., Leirer, V.O.: Development and Validation of a Geriatric Depression Screening Scale: a Preliminary Report. Journal of Psychiatric Research 17, 37–49 (1983)
21. Heerink, M., Kröse, B.J.A., Evers, V., Wielinga, B.J.: Assessing Acceptance of Assistive Social Agent Technology by Older Adults: the Almere Model. I. J. Social Robotics 2, 361–375 (2010)
22. Cortellessa, G., Scopelliti, M., Tiberio, L., Koch Svedberg, G., Loutfi, A., Pecora, F.: A Cross-Cultural Evaluation of Domestic Assistive Robots. In: Proceedings of AAAI Fall Symposium on AI in Eldercare: New Solutions to Old Problems (2008)

23. Steinfeld, A., Lewis, M., Fong, T., Scholtz, J.: Common Metrics for Human-Robot Interaction. In: Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction, pp. 33–40. ACM (2006)
24. Tsai, T.C., Hsu, Y.L., Ma, A.I., King, T., Wu, C.H.: Developing a Telepresence Robot for Interpersonal Communication with the Elderly in a Home Environment. Telemedicine and e-Health 13, 407–424 (2007)
25. Bensalem, S., de Silva, L., Gallien, M., Ingrand, F., Yan, R.: "Rock Solid" Software: A Verifiable and Correct-by-Construction Controller for Rover and Spacecraft Functional Levels. In: Proc. of the 10th Int. Symp. on Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS 2010 (2010)

Author Index