Studies in Computational Intelligence 570

David Camacho
Lars Braubach
Salvatore Venticinque
Costin Badica *Editors*

Intelligent Distributed Computing VIII

Springer

Studies in Computational Intelligence

Volume 570

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

About this Series

The series "Studies in Computational Intelligence" (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution, which enable both wide and rapid dissemination of research output.

More information about this series at http://www.springer.com/series/7092

David Camacho · Lars Braubach
Salvatore Venticinque · Costin Badica
Editors

Intelligent Distributed Computing VIII

Springer

Editors
David Camacho
Computer Science Department
Technical School of Engineering
Universidad Autónoma de Madrid
Madrid
Spain

Lars Braubach
University of Hamburg
Hamburg
Germany

Salvatore Venticinque
Department of Industrial and Information Engineering
Second University of Naples
Aversa
Italy

Costin Badica
Faculty of Automatics, Computers and Ele
Software Engineering Department
University of Craiova
Craiova
Romania

ISSN 1860-949X ISSN 1860-9503 (electronic)
ISBN 978-3-319-10421-8 ISBN 978-3-319-10422-5 (eBook)
DOI 10.1007/978-3-319-10422-5

Library of Congress Control Number: 2014947250

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

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

Preface

Intelligent Distributed Computing is an emerging research field that combines intelligent with distributed computing in order to provide foundations and technologies facilitating the construction of intelligent system behavior in distributed environments. Intelligent Computing includes concepts from classical artificial intelligence, computational intelligence and multi-agent systems to game theory. Distributed Computing is concerned with middleare technology and means for developing systems that are composed of collaborating components. Theoretical foundations and practical applications of Intelligent Distributed Computing set the premises for a new generation of intelligent distributed information systems.

Intelligent Distributed Computing – IDC'2014 continues the tradition of the IDC Symposium Series that was started as an initiative of research groups from: (i) *Systems Research Institute, Polish Academy of Sciences in Warsaw, Poland* and (ii) *Software Engineering Department of the University of Craiova, Craiova, Romania*. IDC aims at bringing together researchers and practitioners involved in all aspects of Intelligent Distributed Computing. IDC is interested in works that are relevant for both Distributed Computing and Intelligent Computing, with scientific merit in at least one of these two areas. IDC'2014 was the eigth event in the series and was hosted by the *Applied Intelligence & Data Analysis Group (AIDA)* at *Escuela Politécnica Superior, Universidad Autónoma de Madrid* in *Madrid, Spain* during September 3-5, 2014. IDC'2014 had a special interest in novel architectures, systems and methods that facilitate distributed, parallel and multi-agent biocomputing for solving complex computational and real-life problems.

IDC'2014 was collocated with: (i) Workshop on Cyber Security and Resilience of Large-Scale Systems (WSRL 2014); (ii) Sixth International Workshop on Multi-Agent Systems Technology and Semantics (MASTS 2014).

The material published in this book is divided into four main parts: (i) 23 long contributions of IDC'2014 participants; (ii) 15 short contributions of IDC'2014 participants; and (iii) 6 contributions of WSRL'2014 participants; and (iii) 3 contributions of MASTS'2014 participants.

The response to IDC'2014 call for paper was generous. We received 65 submissions from 17 countries (we counted the country of each coauthor for each

submitted paper). Each submission was carefully reviewed by at least 3 members of the IDC'2014 Program Committee. Acceptance and publication were judged based on the relevance to the symposium themes, clarity of presentation, originality and accuracy of results and proposed solutions. Finally 23 regular papers and 15 short papers were selected for presentation and were included in this volume, resulting in acceptance rates of 35.38 % for regular papers and 35.71 % for short papers.

The 47 contributions published in this book address many topics related to theory and applications of intelligent distributed computing and multi-agent systems, including: adaptive and autonomous distributed systems, agent programming, ambient assisted living systems, business process modeling and verification, cloud computing, coalition formation, decision support systems, distributed optimization and constraint satisfaction, gesture recognition, intelligent energy management in WSNs, intelligent logistics, machine learning, mobile agents, parallel and distributed computational intelligence, parallel evolutionary computing, trust metrics and security, scheduling in distributed heterogenous computing environments, semantic Web service composition, social simulation, and software agents for WSNs.

We would like to thank to Janusz Kacprzyk, editor of *Studies in Computational Intelligence* series and member of the Steering Committee for his continuous support and encouragement for the development of the IDC Symposium Series. We would like to thank to the IDC'2014, WSRL'14 and MASTS'2014 Program Committee members for their work in promoting the event and refereeing submissions and also to all colleagues who submitted papers to IDC'2014, WSRL'14 and MASTS'2014. We deeply appreciate the efforts of our invited speakers Thomas Stützle,and Juan Julián Merelo Guervos and thank them for their interesting lectures. Special thanks also go to organizers of WSRL'14: Massimo Ficco, and MASTS'2014 organizers: Adina Magda Florea, Amal El Fallah Seghrouchni, and John Jules Meyer. Finally, we appreciate the efforts of local organizers on behalf of *AIDA Group* and *Escuela Politécnica Superior, Universidad Autónoma de Madrid* in *Madrid, Spain* for hosting and organizing IDC'2014, WSRL'14 and MASTS'2014.

Madrid, Hamburg, Naples, Craiova
July 2014

David Camacho
Lars Braubach
Salvatore Venticinque
Costin Bădică

Organization

Organizers

Applied Intelligence & Data Analysis (AIDA) group
Computer Science Department, Universidad Autónoma de Madrid, Spain
Software Engineering Department, Faculty of Automation, Computers and Electronics, University of Craiova, Romania

Conference Chairs

David Camacho, Universidad Autónoma de Madrid, Spain

Steering Committee

Janusz Kacprzyk, Polish Academy of Sciences, Poland
Costin Bădică, University of Craiova, Romania
David Camacho, Universidad Autónoma de Madrid, Spain
Filip Zavoral, Charles University Prague, Czech Republic
Frances Brazier, Delft University of Technology, The Netherlands
George A. Papadopoulos, University of Cyprus, Cyprus
Giancarlo Fortino, University of Calabria, Italy
Kees Nieuwenhuis, Thales Research & Technology, The Netherlands
Marcin Paprzycki, Polish Academy of Sciences, Poland
Michele Malgeri, University of Catania, Italia
Mohammad Essaaidi, Abdelmalek Essaadi University in Tetuan, Morocco

Invited Speakers

Thomas Stützle, Université Libre de Bruxelles, Belgium
Juan Julián Merelo, University of Granada, Spain

Program Committee Chairs

David Camacho, Universidad Autónoma de Madrid, Spain
Lars Braubach, University of Hamburg, Germany
Salvatore Venticinque, Second University of Naples, Italy
Costin Bădică, University of Craiova, Romania

Program Committee

Aquilino A. Juan, Universidad de Oviedo
Salvador Abreu, Universidade de Evora and CENTRIA
Ricardo Aler, Universidad Carlos III de Madrid
Amparo Alonso Betanzos, Universidad Da Coruña
Razvan Andonie, Central Washington University
Mert Bal, Assistant Professor at the Miami University
Nick Bassiliades, Aristotle University of Thessaloniki
David Bednarek, Charles University Prague
Nik Bessis, University of Derby
Lars Braubach, University of Hamburg
Giacomo Cabri, Universita di Modena e Reggio Emilia
Vincenza Carchiolo, Universita di Catania
Jen-Yao Chung, IBM T. J. Watson Research Center
Dorian Cojocaru, University of Craiova
Phan Cong-Vinh, NTT University
Juan Manuel Corchado, University of Salamanca
Alfredo Cuzzocrea, ICAR-CNR and University of Calabria
Dumitru Dan Burdescu, University of Craiova
Paul Davidsson, Malmo University
Javier Del Ser Lorente, Tecnalia Research & Innovation, Bizkaia, Spain
Giuseppe Di Fatta, University of Reading
Amal El Fallah Seghrouchni, LIP6 - University of Pierre and Marie Curie
George Eleftherakis, The University of Sheffield International Faculty, CITY College
Vadim Ermolayev, Zaporozhye National Univ.
Fernando Esteban Barril Otero, University of Kent
Mostafa Ezziyyani, Faculty of Sciences and Technologies of Tanger

David Fernandez Barrero, University of Alcalá
Camino Fernández Llamas, Universidad de León
Giancarlo Fortino, University of Calabria
Stefano Galzarano, University of Calabria (UNICAL), Italy
Maria Ganzha, University of Gdansk, Poland
Marie-Pierre Gleizes, IRIT
Jorge Gomez-Sanz, Universidad Complutense de Madrid
Bertha Guijarro, Universidad Da Coruña
Marjan Gusev, Univ. Sts Cyril and Methodius
Mirjana Ivanovic, University of Novi Sad
Jason J. Jung, Yeungnam University
Francisco Jurado, Universidad Autónoma de Madrid
Igor Kotenko, SPIIRAS
Dariusz Krol, Wroclaw University of Technology, Institute of Applied Informatics
Barna Laszlo Iantovics, Petru Maior University of Tg. Mures
Florin Leon, Technical University "Gheorghe Asachi" of Iasi
Antonio Liotta, Eindhoven University of Technology
Alessandro Longheu, DIEEI - University of Catania
Jose Machado, Universidade Minho
Giuseppe Mangioni, University of Catania
Viviana Mascardi, CS Department (DISI) - Universit degli Studi di Genova
Vicente Matellán, Universidad de León
Grzegorz J. Nalepa, AGH University of Science and Technology
Paulo Novais, University of Minho
Andrea Omicini, Universita di Bologna
Mihaela Oprea, University Petroleum-Gas of Ploiesti, Dept. of Informatics
Agustín Orfila, Universidad Carlos III de Madrid
Carlos Palau, Universidad Politécnica de Valencia
Marcin Paprzycki, IBS PAN and WSM
Juan Pavon, Universidad Complutense Madrid
Stefan-Gheorghe Pentiuc, University Stefan cel Mare Suceava
Dana Petcu, West University of Timisoara
Florin Pop, University Politehnica of Bucharest
Radu-Emil Precup, Politehnica University of Timisoara
Shahram Rahimi, Southern Illinois University
María Dolores Rodríguez Moreno, University of Alcalá
Domenico Rosaci, DIMET Department, University Mediterranea of Reggio Calabria
Emilio S. Corchado, University of Salamanca
Sancho Salcedo Sanz, University of Alcalá
Ioan Salomie, Technical University of Cluj-Napoca
Corrado Santoro, University of Catania - Dipartimento di Matematica e Informatica
Heitor Silverio Lopes, UTFPR
Safeeullah Soomro, Indus University Pakistan
Giandomenico Spezzano, CNR-ICAR and University of Calabria

Stanimir Stoyanov, University of Plovdiv
Juan Tapiador, Universidad Carlos III de Madrid
Rainer Unland, University of Duisburg-Essen, ICB
José M. Valls, Universidad Carlos III de Madrid
Salvatore Venticinque, Seconda Universita di Napoli
Lucian Vintan, Lucian Blaga University of Sibiu
Martijn Warnier, Delft University of Technology
Jakub Yaghob, Charles University Prague
Filip Zavoral, Charles University Prague

Organizing Committee

Gema Bello Orgaz, Universidad Autónoma de Madrid
David Fernandez Barrero, University of Alcalá
Antonio Gonzalez-Pardo, Universidad Autónoma de Madrid
Sorin Ille, University of Craiova
Hector Menendez, Universidad Autónoma de Madrid
Fernando Palero Molina, Universidad Autónoma de Madrid
Cristian Ramirez Atencia, Universidad Autónoma de Madrid
Francisco Rodríguez Moreno, Universidad Autónoma de Madrid

MASTS'2014 Workshop Chairs

Adina Magda Florea, University "Politehnica" of Bucharest, Romania
Amal El Fallah Seghrouchni, Université Pierre & Marie Curie, France
John Jules Meyer, Universiteit Utrecht, The Netherlands

MASTS'2014 Workshop Program Committee

Adina Magda Florea, University Politehnica of Bucharest, Romania
Amal El Fallah Seghrouchni, Université Pierre & Marie Curie, France
Andrei Olaru, University Politehnica of Bucharest, Romania
Andrei-Horia Mogos, University Politehnica of Bucharest, Romania
Antoine Zimmermann, ENS des Mines Saint-Etienne, France
Costin Badica, University of Craiova, Romania
Gerard Vreeswijk, Utrecht University, The Netherlands
John Jules Meyer, Utrecht University, The Netherlands
Irina Mocanu, University Politehnica of Bucharest, Romania
Laurent Vercouter, INSA Rouen, France

Marcin Paprzycki, Polish Academy of Science, Poland
Mehdi Dastani, Utrecht University, The Netherlands
Olivier Boissier, ENS des Mines Saint-Etienne, France
Stefan Trausan-Matu, University Politehnica of Bucharest, Romania
Viorel Negru, West University of Timisoara, Romania

WSRL'2014 Workshop Chairs

Massimo Ficco, Second University of Naples, Italy
Amal El Fallah Seghrouchni, Université Pierre & Marie Curie, France
John Jules Meyer, Universiteit Utrecht, The Netherlands

WSRL'2014 Workshop Steering Committee

Cinque Marcello, University of Federico II, Italy
Boada Germán Santos, Technical University of Catalonia, Spain
Castiglione Aniello, Universitá degli Studi di Salerno, Italy
Ilsun You, Korean Bible University, Korea
Palmieri Francesco, Second University of Naples, Italy
Ricciardi Sergio, Technical University of Catalonia, Spain
Giannelli Carlo, Universitá di Bologna, Italy
Florin Fortis, West University of Timisoara, Romania

Contents

Part I
Invited Papers

Low or No Cost Distributed Evolutionary Computation

Juan Julián Merelo

Abstract. From the era of big science we have back to the "do it yourself" era of science, where you don't have any money to buy clusters and subscribe to grids but still have algorithms that crave many computing nodes and need them for scalability. Fortunately, this coincides with the era of big data, cloud computing, and browsers including JavaScript virtual machines. This talk will concentrate on two different aspects of volunteer or freeriding computing: first, the pragmatic: where to find those resources, which can be used, what kind of support you have to give them; and then, the theoretical: how algorithms can be adapted to a environment in which nodes come and go, have different computing capabilites and operate in complete asynchrony of each other.

1 What Is the Point of Low or No Cost Evolutionary Algorithms

The world has computational resources in spades. Most of them do not belong to you or your lab. That does not mean you cannot use it. The problem is how.

Most theory in parallel computing has been devoted to predict and optimize the performance where the number of nodes, their connections, and the time every one is devoting to the computation is known in advance. However, even if Big Science is not over, the era of Citizen science has started (with SETI@home and then BOINC) and it offers a vast amount of computational resources to attract, if only you know how. But there is a challenge: knowing, or at least having a ballpark, of how your algorithm is going to perform in this uncertain environment, where none of the factors is known: neither the number of nodes, through how they are connected, to how long are they going to be focused on doing what you want them to.

Juan Julián Merelo
Dept. of Computer Architecture and Technology and CITIC-UGR, University of Granada, Spain
e-mail: jmerelo@geneura.ugr.es

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_1

Besides, since Amazon started selling EC2 several years ago, reliable and scalable computing resources are also available for a low price and on demand. Recently, Google has also refurbished its offering lowering their prices. This means that the conjunction of free or low-cost cloud computing engines, volunteer computing systems and and untapped capability of desktop systems can be used for creating massive, or at least potentially massive, distributed computing experiments.

2 Conclusion

In this talk we will offer our experience on using browser-based computing since 1999 [1] and other emerging paradigms, such as peer to peer based computing [2], mainly using evolutionary algorithms.

There are many challenges involved in using these resources: from adapting current algorithms so that they match this environment [3] to check which EA configuration works the best in it, or creating a framework that can use it easily [4]. But the main challenge is that asking people to contribute resources implies that you are opening your science to society and you have to give something in return: you have to adopt a set of best practices that have come to be known as Open Science, because "Give, and it shall be given unto you", you will get as much back from society as you give to it opening your science and explaining it to the public. This, among other things, means that popularity will become directly performance of the metacomputer you create by attracting more users.

References

1. González, J., Merelo-Guervós, J.-J., Castillo, P.A., Rivas, V., Romero, G., Prieto, A.: Optimized web newspaper layout using simulated annealing. In: Mira, J., Sánchez-Andrés, J.V. (eds.) IWANN 1999. LNCS, vol. 1607, pp. 759–768. Springer, Heidelberg (1999)
2. Laredo, J.L.J., Eiben, A.E., van Steen, M., Guervós, J.J.M.: EvAg: a scalable peer-to-peer evolutionary algorithm. Genetic Programming and Evolvable Machines 11(2), 227–246 (2010)
3. Merelo, J.J., Castillo, P.A., Laredo, J.L.J., Mora, A., Prieto, A.: Asynchronous distributed genetic algorithms with JavaScript and JSON. In: WCCI 2008 Proceedings, pp. 1372–1379. IEEE Press (2008)
4. Guervós, J.J.M.: NodEO, a evolutionary algorithm library in Node. Technical report, GeNeura group (March 2014), `http://figshare.com/articles/nodeo/972892`

Automated Algorithm Configuration: Advances and Prospects

Thomas Stützle

Abstract. The design and configuration of optimization algorithms for computationally hard problems is a time-consuming and difficult task. This is mainly This is in large part due to a number of aggravating circumstances such as the NP-hardness of most of the problems to be solved, the difficulty of algorithm analysis due to stochasticity and heuristic biases, and the large number of degrees of freedom in defining and selecting algorithmic components and settings of numerical parameters. Over the recent years, the development of automatic methods to search large configuration spaces has received significant attention as a possible solution to these challenges. Such automatic algorithm configuration methods have by now proved to be instrumental for developing high-performance algorithms.

The presentation will discuss how automatic algorithm configuration tools can be used to develop high-performing evolutionary and other optimization algorithms. After an overview of available tools, I will highlight various successful applications of these such as the automatic configuration of multi-objective optimizers, and the automatic configuration of hybrid stochastic local search algorithms. Finally, I will highlight the impact automatic algorithm configuration has and will have on the algorithm design and development process.

Thomas Stützle
Universit Libre de Bruxelles (ULB), Belgium
e-mail: stuetzle@ulb.ac.be

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_2

Part II
Affective Computing

Statistical Inference for Intelligent Lighting: A Pilot Study

Aravind Kota Gopalakrishna, Tanir Ozcelebi, Antonio Liotta, and Johan J. Lukkien

Abstract. The decision process in the design and implementation of intelligent lighting applications benefits from insights about the data collected and a deep understanding of the relations among its variables. Data analysis using machine learning allows discovery of knowledge for predictive purposes. In this paper, we analyze a dataset collected on a pilot intelligent lighting application (the *breakout* dataset) using a supervised machine learning based approach. The performance of the learning algorithms is evaluated using two metrics: *Classification Accuracy* (CA) and *Relevance Score* (RS). We find that the breakout dataset has a predominant *one-to-many* relationship, i.e. a given input may have more than one possible output and that RS is an appropriate metric as opposed to the commonly used CA.

1 Introduction

Many intelligent applications such as the next-generation networked systems [10] involve collecting data from different sources, organizing them and then perform data analysis. Data analysis here refers to the process of acquiring information from the collected data and making conclusions and decisions based on these that are useful for the application. Data analysis depends on the type of the data collected, either quantitative or qualitative [1]. Quantitative data can be analyzed using statistical operations such as frequency distributions, central tendency (using mean,

Aravind Kota Gopalakrishna · Tanir Ozcelebi · Johan J. Lukkien
System Architecture and Networking (SAN), Department of Mathematics and Computer Science, Intelligent Lighting Institute (ILI), Eindhoven University of Technology, The Netherlands
e-mail: {a.kota.gopalakrishna,t.ozcelebi,j.j.lukkien}@tue.nl

Antonio Liotta
Electro-Optical Communications (ECO), Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands
e-mail: a.liotta@tue.nl

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_3

median and mode), correlation and regression. Analyzing qualitative data involves examining the data collected through surveys, interviews and observations. Data analysis helps to understand the relationship between input and output, thereby enabling to make decisions upon designing and implementing desired applications. In this paper, we analyze the data collected for the purpose of developing an intelligent lighting application, where users are dynamically presented with the light settings of their preferences in various contexts.

Intelligent lighting [6] is an application that makes use of contextual information such as user identity, type of activity, influence of external light, time of the day and more to provide a suitable lighting to its users. The pilot setup for intelligent lighting is a particular part of an office space, known as the *breakout* area. A breakout area is an area where office employees can have informal meetings or some time for personal retreat. The breakout area implementation for intelligent lighting [6] contains numerous connected lighting elements and sensors. The challenge here is to develop an intelligent application that learns from the data collected and uses this knowledge in the future to predict a suitable light setting for a given scenario. To design such a system, it is necessary to understand how the input parameters and output light settings are related. Machine learning is a data analysis technique that can be used to discover knowledge from the data for predictive purposes such as intelligent lighting.

In this paper, we investigate the nature of the breakout dataset and present the insights gained into the properties that enables to make better design decisions towards implementing intelligent lighting. The pilot setup of the breakout area is discussed. Subsequently, the details of the breakout dataset and how the data are collected and organized into the breakout dataset are explained. The dataset is processed using supervised machine learning algorithms. The prediction performance is evaluated using two metrics: *Classification Accuracy* (CA) and *Relevance Score* (RS) [7]. CA measures how precise the light prediction is, given a certain environment state. In contrast, RS gives the measure of how relevant a light prediction is for a given state of the environment. Analysis of the results show that the breakout dataset has one-to-many relationships, i.e. a given input (i.e. a context or a state of the environment) may have many possible output light settings. Furthermore, when it comes to intelligent lighting applications, RS is more appropriate than the common performance metric CA.

The paper is organized as follows. In Section 2, the pilot setup of the breakout area, the description of the breakout dataset such as the number of samples, output class distribution and user-sample distribution, and the means in which data are collected and processed are discussed. The experiments performed and the insights from the results are discussed in Section 3. The paper is concluded in Section 4.

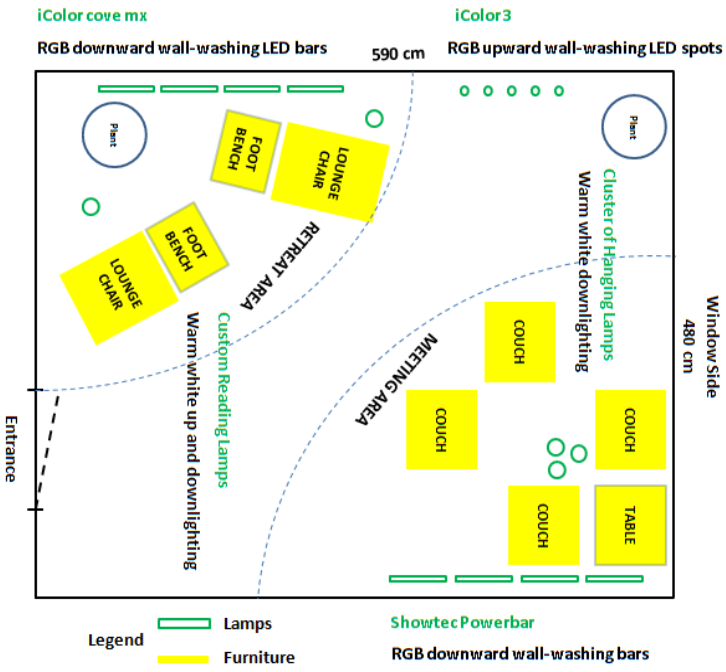


Fig. 1 Floor plan of the breakout area

2 Pilot Setup and Data Collection

In this section, we discuss the pilot implementation setup of the breakout area where the data have been collected, provide a description of the breakout dataset and explain process in which data have been collected.

2.1 Pilot Setup

Figure 1 shows the floor plan of the breakout area. Opposite to the entrance is a wall with windows and blinds, which allows for controlling the external light influence. Furthermore, the area is divided into two spaces dedicated to different purposes: meeting area for informal meetings and the retreat area for personal retreat and relaxation. However, the users of the breakout area are not restricted to use a specific area for a specific activity. For example, user A may choose to use either the meeting area or the retreat area for *relaxation*. Given an intelligent lighting application, the desired light settings in an area may depend on user identity, type of activity in the area and external light influence (sunlight), an area the user may choose for the activity and many more features. The lighting system in the breakout area

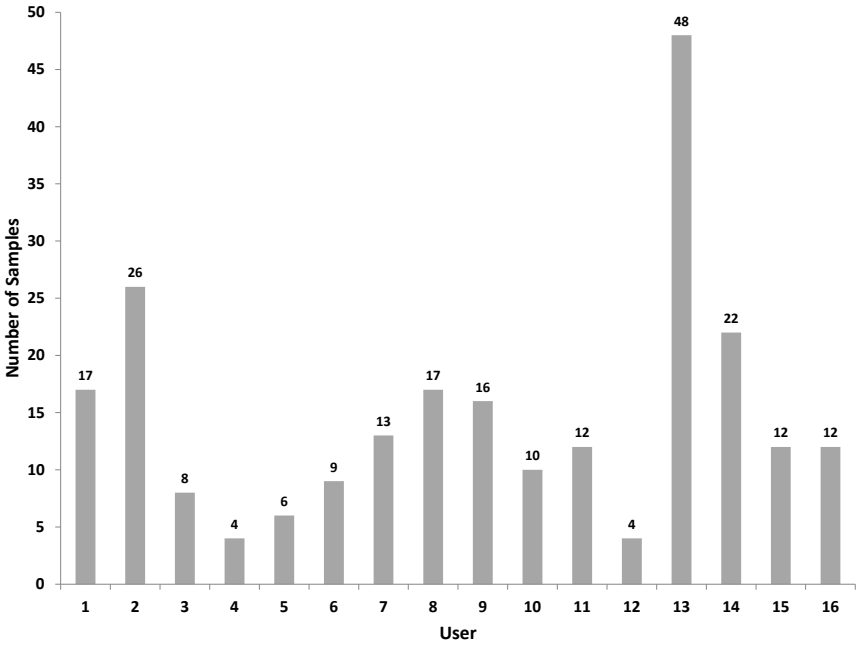


Fig. 2 The distribution of samples in the breakout dataset

contains two types of lights: colored wall-washing lights for creating an atmosphere and white *down-lights* [12] for illuminating areas of user tasks. The sensor system for monitoring the breakout area contains Passive Infra Red (PIR) sensors for monitoring movements, sound pressure sensors for monitoring sound volume intensity and light sensors for measuring external light influence.

Table 1 List of input features considered, that influences user's choice of light selection

	Feature	Type of the feature	Possible Values
1.	User-Identity (UID)	Categorical	U1, U2, U3, U4, ...
2.	Type of Activity (ToA)	Categorical	Active_Group, Active_Alone, Relax_Group, Relax_Alone
3.	Area of Activity (AoA)	Categorical	Meeting, Retreat
4.	Intensity of Activity (IoA) in the other subarea	Categorical	$0, 1, 2, \ldots, 10$
5.	Time of the Day (ToD)	Numeric	$\in [0, 24)$, e.g. 10.5 for 10:30am
6.	External Light Influence (ExLI)	Categorical	VeryHigh, High, Low, VeryLow

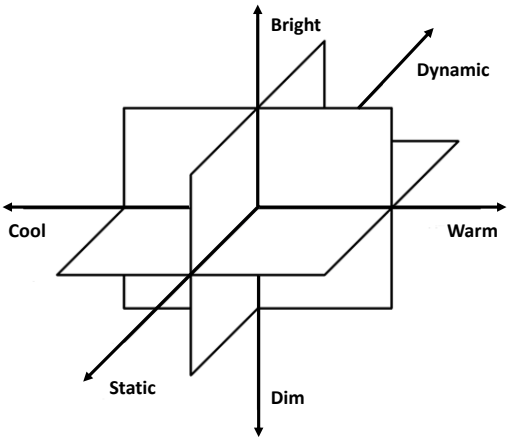


Fig. 3 Possible Output Light Combinations

2.2 Description of the Breakout Dataset

The breakout dataset for intelligent lighting consists of 236 samples collected as discussed in Section 2.3. The dataset does not have any missing data. We select six input features that may influence a user's choice in selecting one of the pre-defined light settings for a given context as summarized in Table 1. Figure 2 shows the number of users and the numbers of data samples collected per user. We consider eight output light settings to support users' activities as shown in Fig. 3. The class distribution of these eight light settings over the 236 samples is presented in Fig. 4.

2.3 Data Collection

Among the mentioned features in Table 1, AoA, ToD, IoA and ExLi are gathered implicitly from the breakout area through sensor monitoring. The features UID and ToA are acquired explicitly from the users via the breakout application installed on their smart phones.

The data samples for the breakout dataset were collected using two methods. In the first method, we created various contexts in the breakout area with different ExLI, IoU values, in which the participants were asked to select a light setting that they prefer for the activities listed in Table 1. In the second method, the participants were allowed to use the breakout area on-demand for six weeks. During this six-week period, all interactions of the users with the system (i.e. activities and selected light settings) as well as the sensor readings were logged.

In order to learn users' preferences of light settings in a particular context, collected data samples should contain the values for features UID and ToA. On the

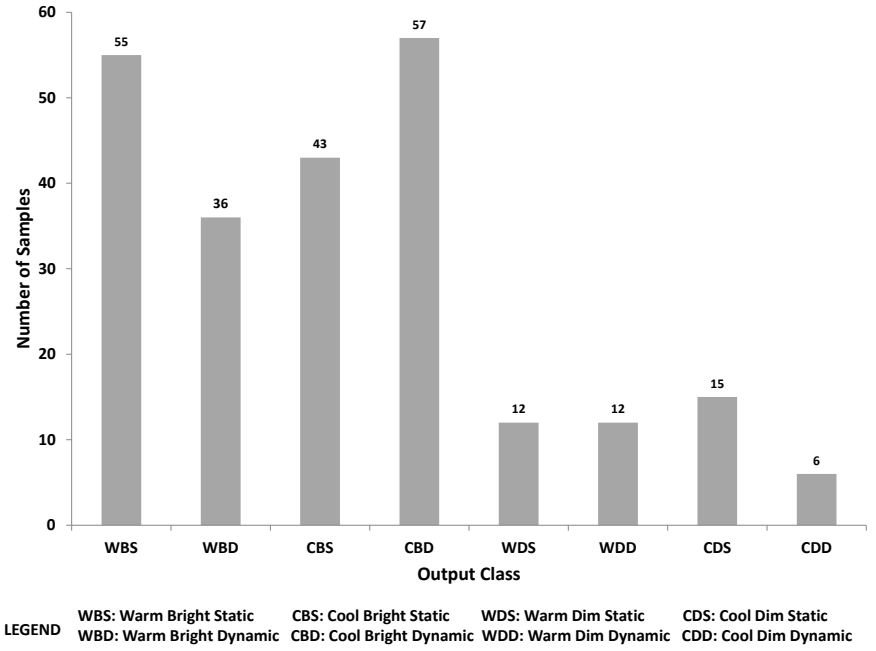


Fig. 4 The distribution of output class in the breakout dataset

other hand, when there are no users in the breakout area, then the data samples collected contain no entries for these features. The breakout dataset is then obtained by performing data cleaning in two steps. Firstly, those samples that do not contain feature values for UID and ToA are filtered out. Secondly, those samples that belong to users' free explorations of different light settings for a particular context are removed.

3 Experiments and Discussion

We use supervised learning algorithms to analyze the breakout dataset by investigating the prediction performance using two different metrics: *Classification Accuracy* (CA) and *Relevance Score* (RS). The following six rule-based prediction models in WEKA [8] are considered: DecisionTable [9], JRip [3], Nearest Neighbor with generalization (NNge) [11], PART [4], ConjunctiveRule [2] and Ridor [5]. The prediction performance of the prediction models on the breakout dataset are computed using 10-fold cross-validation.

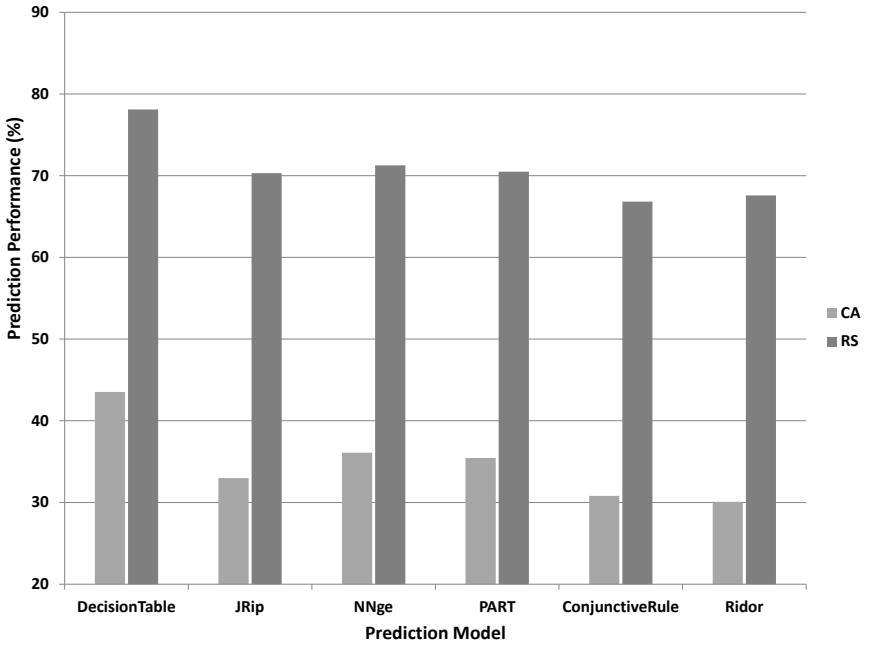


Fig. 5 Prediction performance vs. Prediction model for six rule-based prediction models with 8-output light settings

3.1 Analysis of Prediction Performance with 8-Outputs

Figure 5 presents the prediction performance of the six rule based classification models considering CA and RS as metrics. It can be seen that CA values are very low for all the considered prediction models, compared to RS values. This is because the CA metric measures how accurate the prediction is for a sample, i.e. the predicted outcome is compared to the actual outcome. If the predicted and actual outcomes do not match, then the CA metric scores a zero. Since users are not consistent in selecting a particular light setting for a given context, the average CA for a lighting application is typically low. The inconsistency comes from the fact that it is very difficult (indeed impossible) to consider the full set of input features (context) that determine a user's light setting choice. Furthermore, some contextual information, such as a user's mood, can not be monitored easily. Instead, a learning algorithm takes only a part of all relevant input features (i.e. an observed context) into account. Since, multiple light settings can satisfy a user in a given observed context, the nature of the breakout dataset i.e. the input-output relationship is *one-to-many*. The RS metric measures how relevant the predicted outcome is, for a given context based on the information computed from the dataset [7]. The RS metric does not score a zero when there is a mismatch between the predicted and actual outcome and thus gives higher performance.

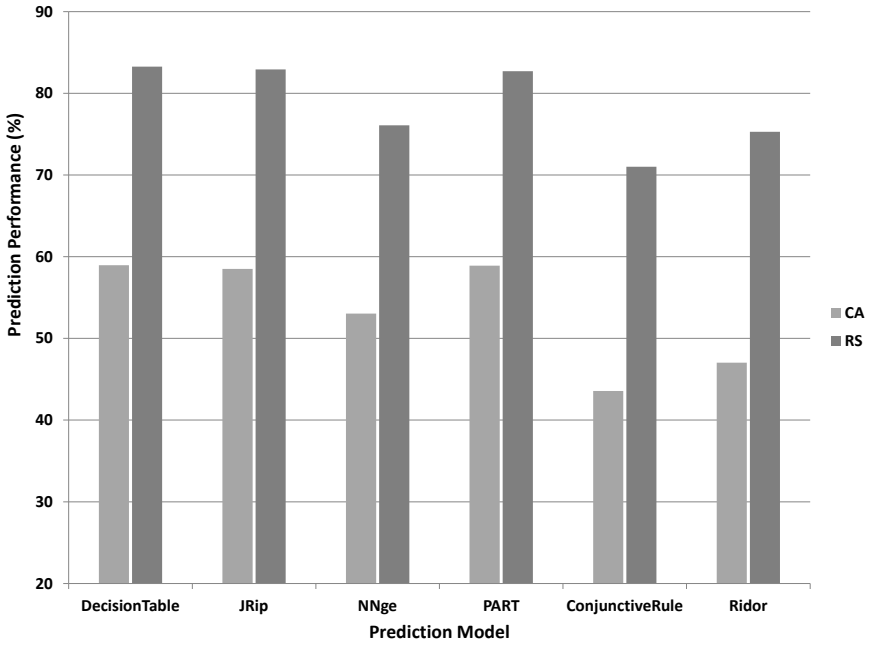


Fig. 6 Prediction performance vs. Prediction model for six rule-based prediction models with 4-output light settings

3.2 Analysis of Prediction Performance with 4-Outputs

In this experiment, static and dynamic light settings are combined into 4-output classes. This is done because the users could not differentiate much between the static and dynamic settings [12]. Figure 6 presents the performance values of the six rule based classification models considering CA and RS as metrics. It can be seen that since the output space is reduced, the CA values improve as compared to the results of the 8-output dataset. However, the RS values do not improve much.

3.3 On Implementing Intelligent Lighting

Table 2 shows the standard deviation of the prediction performance for the six prediction models computed using 10-fold cross-validation. The values show that there is a high degree of inconsistency in the prediction performance for both the metrics considered. This means that the performance of the prediction models that use supervised learning approach varies significantly with different training and test sets.

From this study, we find that the use of supervised learning algorithms for implementing intelligent lighting with a metric such as CA is inappropriate considering

the nature of the breakout dataset. The RS metric is better as it evaluates the prediction performance from a different perspective. Furthermore, the selection of the learning approach to implement intelligent lighting depends on the objectives to be achieved such as whether the system should learn and adapt continuously or not. Supervised learning algorithms are trained on a fixed dataset. These algorithms does not adapt as the input-output relationships change in time due to dynamic factors such as changing user preferences and changing lengths of daytime in different seasons. Therefore, it is necessary to explore other learning techniques such as online learning and analyze their prediction performance.

Table 2 Standard deviation of the prediction performance for the six rule-based prediction models

Prediction Model	Std Dev (CA)	Std Dev (RS)
Decision Table	7.78	5.45
JRip	7.80	4.74
NNge	8.55	5.90
ConjunctiveRule	8.48	5.23
PART	6.88	7.77
Ridor	11.38	8.08

4 Conclusion

Data analysis of an intelligent lighting application leads to insights into the relations among various input features as well as suitability of different performance metrics and performance limitations. In designing such applications, such insights help in deciding upon a certain sensor modality and learning algorithm. By means of statistical analysis of a dataset collected from a pilot implementation named the breakout area, we were able to infer that the breakout dataset has a *one-to-many* input-output relationship unlike most available real-world datasets. This means that more than one output may be satisfying for a given input context. The experiments were performed using six rule-based prediction models and two performance evaluation metrics: Classification Accuracy (CA) and Relevance Score (RS). We find that the CA is not an appropriate metric for applications such as intelligent lighting having *one-to-many* input-output relationship and that the RS is most appropriate performance metric. As a future work, we will investigate other learning techniques such as online learning and reinforcement learning and analyze their prediction performance.

Acknowledgements. This work has been supported under the Smart Context Aware Services project (SmaCS) through the Point One grant No. 10012072. We would also like to thank Serge Offermans, Intelligent Lighting Institute (ILI), Eindhoven University of Technology for the pilot implementation of the *Breakout* area.

References

1. Analyzing and Interpreting Data. Technical report, Wilder Research (2009)
2. Clarke, P., Niblett, T.: The CN2 Rule Induction Algorithm. In: Machine Learning, pp. 261–283 (1989)
3. Cohen, W.H.: Fast Effective Rule Induction. In: Proceedings of the Twelfth International Conference on Machine Learning, pp. 115–123. Morgan Kaufmann (1995)
4. Frank, E., Witten, I.H.: Generating Accurate Rule Sets Without Global Optimization. In: Proceedings of the Fifteenth International Conference on Machine Learning, pp. 144–151 (1998)
5. Gaines, B.R., Compton, P.: Induction of ripple-down rules applied to modeling large databases. Journal of Intelligent Information Systems 5(3), 211–228 (1995)
6. Gopalakrishna, A.K., Ozcelebi, T., Liotta, A., Lukkien, J.J.: Exploiting Machine Learning for Intelligent Room Lighting Applications. In: Proceedings of the 6th IEEE International Conference on Intelligent Systems, pp. 406–411 (2012)
7. Gopalakrishna, A.K., Ozcelebi, T., Liotta, A., Lukkien, J.J.: Relevance as a Metric for Evaluating Machine Learning Algorithms. In: Perner, P. (ed.) MLDM 2013. LNCS (LNAI), vol. 7988, pp. 195–208. Springer, Heidelberg (2013)
8. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1) (2009)
9. Kohavi, R.: The Power of Decision Tables. In: Lavrač, N., Wrobel, S. (eds.) ECML 1995. LNCS, vol. 912, pp. 174–189. Springer, Heidelberg (1995)
10. Liotta, A.: The Cognitive Net is Coming. IEEE Spectrum 50(8), 26–31 (2013)
11. Martin, B.: Instance-based Learning: Nearest Neighbor with Generalization. Technical Report, University of Waikato (1995)
12. Offermans, S., van Essen, H., Eggen, B.: Exploring a Hybrid Control Approach for enhanced User Experience of Interactive Lighting. In: Proceedings of the 27th International BCS Human Computer Interaction Conference, pp. 1–9 (2013)

How Musical Selection Impacts the Performance of the Interaction with the Computer

Mickael da Costa, Davide Carneiro, Marcelo Dias, and Paulo Novais

Abstract. In this busy society of ours people push their limits to work better and more in order to remain competitive with their peers. Nonetheless, working longer hours does not necessarily improves productivity nor performance. In order to prevent the negative consequences of this increasing trend, the evolution of performance throughout the day of work should be more closely monitored. This could avoid undesirable states or even breakdowns, which have social and economical implications. In this work we measure user performance through their interaction with the computer. We monitor its evolution during a day of work and how different types of music may increase or decrease its natural daily degradation. We conclude that the relationship between types of music and its effects is not universal and depends, among other things, on the musical profile of the individual. A prototype for a distributed music recommendation service is presented that suggests musics at an individual and group level, based on user musical profiles and objectives.

Keywords: Music, Performance, Context Acquisition, Human-Computer Interaction.

1 Introduction

The Human being is currently under an increasing demand for performance, fruit of a competitive society in which the scarcity of resources drives individuals into harsher conditions. Workplaces are particularly "good" examples of this reality.

Mickael da Costa · Davide Carneiro · Paulo Novais
CCTC/DI, Universidade do Minho, Braga, Portugal
e-mail: dinomickael@gmail.com, {dcarneiro,pjon}@di.uminho.pt

Marcelo Dias
Escola de Psicologia, Universidade do Minho, Braga, Portugal
e-mail: marcelofvdias@gmail.com

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_4

Lack of jobs, decreasing wages, increasing working hours, working in shifts, competitiveness or unrealistic productivity goals result in a constant and increasing pressure on the individual.

Numerous studies highlight the negative effects of this lifestyle. [12] show positive mean correlations between overall health symptoms, physiological and psychological health symptoms, and hours of work. [5] analyse the impact of overtime and long work hours on occupational injuries and illnesses, to conclude that these variables depend more on the amount of time worked rather than on the level of risk of the job. In [7], the effects of shift work and extended hours of work are analysed at different levels, including family and social life, performance, fatigue, productivity, health, among others.

As addressed in detail in [2], there is currently an overwork culture, which is further encouraged by greedy management techniques and job insecurity. While the main objective of management is to increase production, this does not necessarily happen, nor will it increase productivity.

There is thus the need to improve performance or productivity by other means that do not bring along such negative effects. This paper presents such an approach on the problem through the use of music. Indeed, musical selection affects many different aspects of our lives, including our physiology, mood or motivation [1]. It is one of the oldest forms of cultural expression and, given its effects on the Human being at so many different levels (e.g. emotional, health, physiological), has been studied in the last decades for many different purposes, including its effects on shopping behavior, its therapeutic possibilities, its effects on sport and exercise and even its effects on our emotions [6, 8, 10, 13].

The strategies for selecting music are often driven by the objective of activating or calming people. In this work we focus on environments in which groups of people work together, mostly in front of a computer, typically offices (e.g. software development, call center, journals).

The aim of this work is to determine the potential effect of music on the natural degradation of performance that occurs during the workday. Specifically, we want to determine if particular types of music can decrease this natural degradation, contributing to a higher overall performance of the individual. This will be done individually when people can make use of headphones, or in group when only sound systems are available. To this end we develop a distributed music recommendation service that, based on user profiles and a real-time analysis of performance, selects the most adequate music. The main innovative point is indeed this real-time analysis of performance, which is based on the analysis of the interaction with the mouse and keyboard. It is therefore non-invasive and requires no conscious or specific interaction from its users, making it suited to be used in permanence in the workplace, as opposed to other existing solutions.

2 Distributed Context Acquisition for Performance Indicators

To implement the study described in this paper and the developed application prototype, a framework for the acquisition of performance indicators from contextual information was developed. This framework is based on a client-server model 1. The clients (the computers used by the individuals in the environment) provide information about their users while the server receives it and builds the set of features that feed the decision-making and presentation tier.

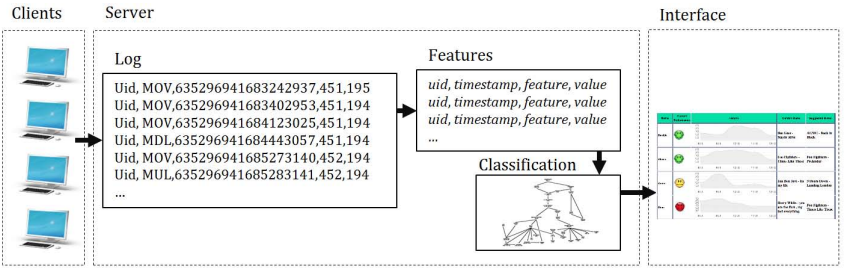


Fig. 1 High-level view of the architecture

The clients use the keyboard and mouse as sensors of user performance. Specifically, the user is monitored through particular Operating System events fired from the use of the computer's mouse and keyboard, as follows:

- MOV, timestamp, posX, posY - an event describing the movement of the mouse, in a given time, to coordinates (posX, posY) in the screen;
- MOUSE_DOWN, timestamp, [Left|Right], posX, posY - this event describes the first half of a click (when the mouse button is pressed down), in a given time. It also describes which of the buttons was pressed (left or right) and the position of the mouse in that instant;
- MOUSE_UP, timestamp, [Left|Right], posX, posY - an event similar to the previous one but describing the second part of the click, when the mouse button is released;
- MOUSE_WHEEL, timestamp, dif - this event describes a mouse wheel scroll of amount dif, in a given time;
- KEY_DOWN, timestamp, key - identifies a given key from the keyboard being pressed down, at a given time;
- KEY_UP, timestamp, key - describes the release of a given key from the keyboard, in a given time;

The clients share these events in real-time with the server, whom proceeds to compute the following set of features: Key Down Time, Time Between Keys, Mouse Velocity, Mouse Acceleration, Time Between Clicks, Double Click Duration, Average Excess of Distance, Average Distance of the Mouse to the Straight Line, Distance of the Mouse to the Straight Line Between two Clicks, Signed Sum of Angles

of the Movement, Absolute Sum of Angles of the Movement and Distance between clicks. These features and the process of their computation are described in more detail in [4].

From these features it is possible to obtain a measure of the user's performance (e.g. an increased distance between clicks or sum of angles represents decreased performance). Then, a wide range of possibilities become real, such as studying the effects of fatigue or stress on performance [4, 11] or, as in this case, the effects of musical selection.

3 Experimental Study

In order to determine the influence of music on the interaction of the individuals with their computers and on their behaviour within the environment, an experimental study was carried out. This study aimed to:

- Determine if musical selection (the independent variable) has an effect on the performance of the interaction patterns (the dependent variable) of the users with the computer;
- Determine if different types of music have different effects on the variable;
- Study and quantify the effects of different types of music on the variable;
- Determine if users are conscious of the effects measured or, at least, of some effect at some level;

In the past we have studied how performance is negatively influenced by fatigue throughout the day. In the present study we aim to determine if music may have a positive effect on the performance of the individual by improving it or by delaying its decrease during the day or during particularly stressing periods. The verification of this possibility will support the development of a music recommendation service aimed at improving musical selection with particular objectives, such as improving individual or group performance, satisfaction with music selection or motivation to work. This is expected to consequently improve indicators such as work satisfaction, productivity and quality of the working environment.

3.1 Method

This experimental study took place in the Intelligent Systems Lab of the University of Minho. In this lab, numerous students and researchers spend their day working with a specific computer and are allowed to listen to music using headphones. 12 participants were selected to take place in this study, aged between 20 and 28, with an average of 24.3.

Prior to the participation in the study, each individual filled in a questionnaire aimed at determining their musical preferences. Moreover, at the end of each day,

they also filled in another questionnaire to determine their subjective opinion about the musical selection of the day.

The selected individuals were requested to participate in the study for five days. During their participation they need not change any of their routines: the only request was that they carried out their usual tasks while listening to the provided music using their headphones.

The recording of their performance indicators was carried out in the background through a log application that required no interaction at all.

The independent variable in this study was thus musical selection. Five different types of music were used, first classified and put forward by [9] in the form of five so-called mood clusters. Each cluster contained music classified as follows:

- Cluster 1: passionate, rousing, confident, boisterous, rowdy
- Cluster 2: rollicking, cheerful, fun, sweet, amiable/good natured
- Cluster 3: literate, poignant, wistful, bittersweet, autumnal, brooding
- Cluster 4: humorous, silly, campy, quirky, whimsical, witty, wry
- Cluster 5: aggressive, fiery, tense/anxious, intense, volatile, visceral

The dependant variable was the performance of the participants, measured in terms of the features described previously.

Before the actual start of the data collection, each participant filled in a first questionnaire, meant to establish a profile of each participant. In this questionnaire, each participant provides some standard demographic data, rates some musics from the different clusters according to their level of activation or valence (from the participant's point of view) and answers some questions that allow to perceive their musical preferences.

During the actual study, each participant took part in five different moments of data collection, each one in a different day. In each day, the participant listens to musics from one of the different clusters during the whole period of work, with a minimum of 3 hours.

At the end of each day, each participant answered another questionnaire aimed to determine how the type of music listened made them feel concerning their performance at work (e.g. is the participant consciously aware of some effect?). Moreover, it was also the aim of this questionnaire to determine if the music truly induced the desired state in the participant.

The data collected, from both the questionnaires and the performance monitoring software, was analysed using statistical software and the results are described further ahead in this paper.

3.2 Results

Given the scope of the paper, we will not dwell to deep into the results of the study: we will only focus on the most important aspects that allow us to grasp the relationship between music and performance.

One of our objectives was to determine if the musical selection in each cluster would be experienced by the participants as expected, i.e., if the clusters we deemed to be calm would be considered calming by the participants. As Figure 2 shows, this happens indeed. Cluster 3, containing music classified as autumnal, brooding or literate, is the one that relaxes participants the most. Cluster 5, on the other hand, containing music described as aggressive, fiery or tense/anxious, is the one that relaxes them the least.

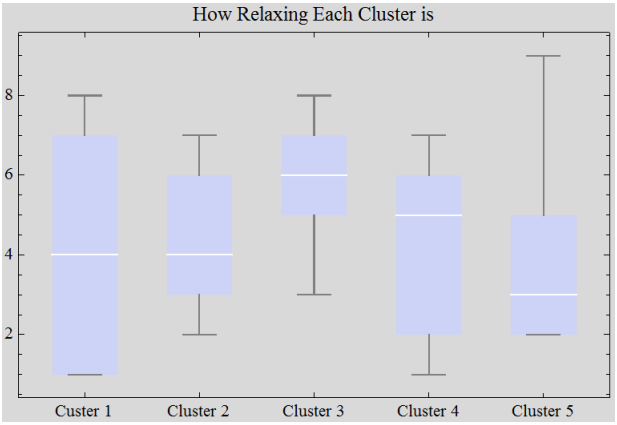


Fig. 2 Distribution of the gradings, by the users, of how relaxing each Cluster is (1 - no relaxing at all, 9 - highly relaxing)

There is not necessarily a direct relationship between how relaxed you are and your performance. Indeed, this relationship is more complex than it may seem at first sight. In this study we found that our performance does not depend only on the musical selection but also on the musical profile of the individual.

Indeed, if we consider Figure 3, we notice that the Cluster that attenuates fatigue the most over the day is Cluster 4, while the ones that contribute to increasing the effects of fatigue the most are Clusters 2 and 5. This can be explained by the fact that Cluster 4 contains music that can be described as humorous and silly, contributing to the good mood and motivation of the participant. Cluster 2, although somewhat similar, is more calm and activates people less. Cluster 5 contains heavy music that, over longer periods, will wear the participants out, producing negative effects. These characteristics can help to understand the observed differences.

These differences, although visible when looking at all the population, can result still more interesting when considering individual participants and their musical profile. Indeed, we observe that people that are more into heavy music are positively affected and see their performance improved by a longer timespan with the heavier clusters. They are activated by this music in a positive manner and work more efficiently. These are also people that have a higher baseline activation, i.e., they are naturally more "stressed". People that are naturally calmer, on the other hand, find this music annoying and sometimes hurtful to hear and are unable to

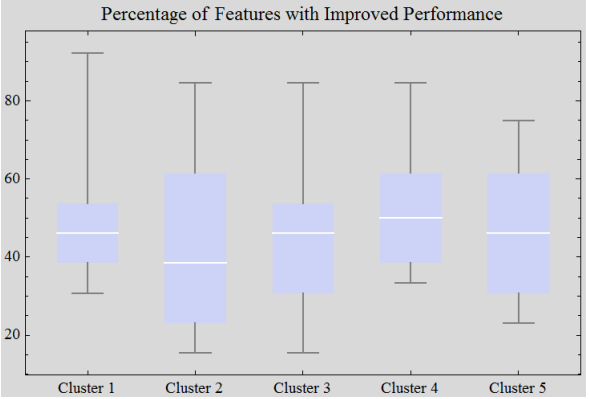


Fig. 3 Distribution of the percentage of features in which each participant improved their performance over the day, for each Cluster

concentrate, which affects their performance. These individuals thus work with more performance with more calm music.

A good example of this are participants "Davide" and "Vitor Neto" (Figure 4). Davide, who can be described as someone who regularly listens to heavy music, achieves the best performance results with Clusters 2 and 5. Vitor, on the other hand, a calmer person by nature, demonstrates better performance in Clusters 3 and 4.

User	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Alexis	53.8462	23.0769	15.3846	38.4615	46.1538
Andre	38.4615	46.1538	53.8462	61.5385	69.2308
Angelo	53.8462	23.0769	84.6154	84.6154	61.5385
Claudia	30.7692	23.0769	46.1538	53.8462	46.1538
Davide	46.1538	84.6154	46.1538	38.4615	61.5385
Angelo	50	41.6667	33.3333	50	75
JosePacheco	61.5385	69.2308	53.8462	38.4615	23.0769
Kevin	41.6667	66.6667	33.3333	33.3333	25
Luis	61.5385	30.7692	69.2308	46.1538	61.5385
Ricardo	92.3077	61.5385	30.7692	61.5385	46.1538
VitorNeto	38.4615	38.4615	61.5385	76.9231	30.7692
Dino	38.4615	15.3846	30.7692	69.2308	30.7692

Fig. 4 Percentage of features that improved over the day, for each user and each cluster

Indeed, the problem of determining the most appropriate style of music for an individual is a complex one and, as these results show, several variables must be taken into account. Namely, and besides the type of music, the musical profile of the individual. Moreover, the objective of the individual at the time (e.g. does he need to complete a task quickly? Does he prefer to work calmly?) as well as the

timespan (e.g. we have the tendency to grow tired of a type of music if listen to it for prolonged periods of time) should also be included in the future.

4 Music Recommendation Service

Based on the results described in the previous section, we started the development of a prototype for a music recommendation service (Figure 4). This prototype has as main objective to select the most appropriate style of music at a given time. It can do so at two different levels: user-level (in which the prototype optimizes the musical selection for a given user) and group-level (in which it does so for a group of people). In both cases, the process is similar. The prototype is a distributed one. Each user is interacting with a particular computer, which is monitoring his performance and client.

From the profiles defined through the questionnaires, we know how each individual feels about each type of music: how much they like it, how relaxing/activating they find it or to which extent they preferred to have carried out their activities without listening to this particular type of music.

The users also provide the prototype with their current objective, using a minimalist interface. At any moment, the objective can be to relax (e.g. when the user is involved in a creative task), to activate (e.g. when the user needs to complete a given task quickly) or, aside from performance issues, to listen to their favourite musics. Besides assigning an objective, the user also assigns a weight to *Performance* and *Musical Preference*. That is, placing a bigger weight on *Performance* will result in musics that contribute more to the user's activation, despite his preferences. On the other hand, musics that are more to the taste of the user will be selected, despite less effective results being expected in what concerns performance.

These variables, as well as the weights assigned by the user, are used by optimization functions to attribute a score to each Cluster, at any moment, normalized in the interval [0,1]. To prevent people from getting tired of constantly listening to the same type of music, musics are then selected from all the five Clusters in a frequency that is proportional to these scores (e.g. if Cluster 1 has twice the score of Cluster 2, musics form Cluster 1 will be selected with twice the probability).

A similar process is used for selecting ambient music. However, in this case, it is the ambient manager that determines the objective of the environment. If there is a scheduled brainstorming session, the manager may decide to put activating music in order to stimulate ideas and actions. On the other hand, if the end of the day is approaching, the manager may decide to put more relaxing music as individuals are already tired and activating music may have negative effects.

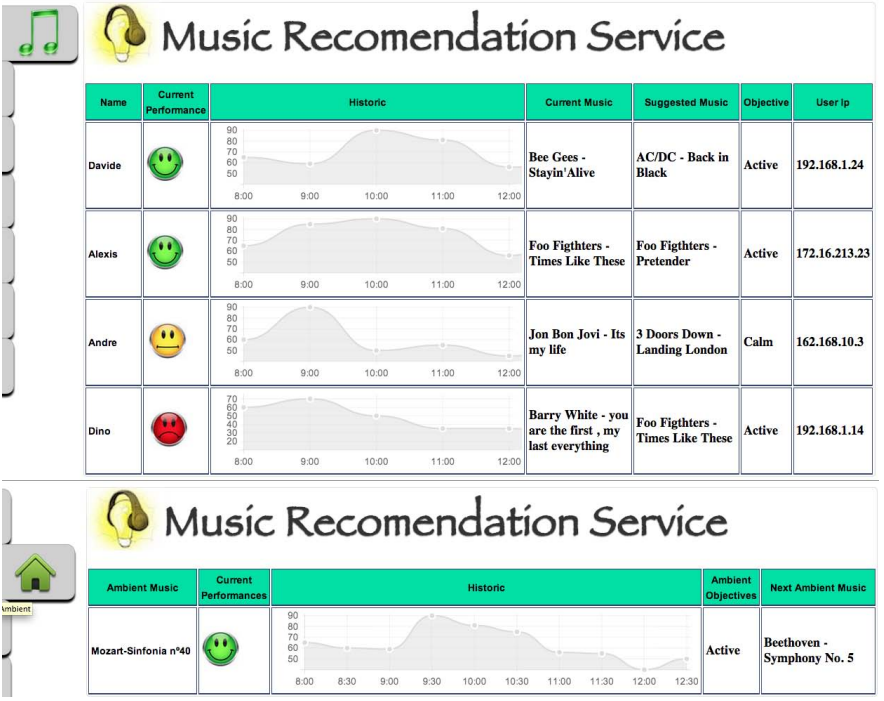


Fig. 5 Detail of the interfaces of the Music Recommendation Service for individual users (upper image) and for the group of users (lower image)

5 Conclusion

Fatigue and its negative effects cause nowadays growing concern. These effects have impact not only at a personal level, wearing one's health, but also at a social level (e.g. our reduced time for social and enjoyable activities) and also at an economical one (companies' costs with absenteeism and reduced productivity are on the rise). Given the current economic scenario, targeting the source of the problem (e.g. decreasing labour time, imposing more favourable legislation) may not be the most realistic or time-efficient solution. In that sense, alternatives should be sought to minimize these negative effects.

In this paper we looked at the possibility of using music to attenuate the negative effects of fatigue on the individual. Specifically, we looked at how performance, measured in terms of the interaction with the mouse and keyboard, decreases along the day and how different types of music affect this phenomena.

We conclude that the relationship is a complex one and involves variables other than the type of music, including the objective of the individual in each moment and his personal preferences regarding music. The data collected in the experimental study was used to define optimization functions that are used to maximize different

aspects of this relationship: to select the favourite musics of the users, to select the music that active the users the most or to select the music that calms them the most.

In future work we will address this problem in more detail, namely by including additional variables that can better shape the relationship between music, performance and fatigue and by analysing different music classification mechanisms.

Acknowledgements. This work is part-funded by ERDF - European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT - FundaÃğÃčo para a CiÃłncia e a Tecnologia (Portuguese Foundation for Science and Technology) within project FCOMP-01-0124-FEDER-028980 (PTDC/EEI-SII/1386/2012).

References

1. Burns, L., Labbé, E., Arke, B., Capeless, K., Cooksey, B., Steadman, A., Gonzales, C.: The effects of different types of music on perceived and physiological measures of stress. Journal of Music Therapy 39(2), 101–116 (2002)
2. Bunting, M.: Willing Slaves: How the Overwork Culture is Ruling Our Lives. Harper Perennial (2005) ISBN 978-0007163724
3. Carneiro, D., Castillo, J.C., Novais, P., Fernández-Caballero, A., Neves, J.: Multimodal behavioral analysis for non-invasive stress detection. Expert Systems with Applications 39(18), 13376–13389 (2012)
4. Carneiro, D.: An Agent-based Architecture for Online Dispute Resolution. PhD Thesis, `http://repositorium.sdum.uminho.pt/handle/1822/28773` (accessed in April 2014)
5. Dembe, A.E., Erickson, J.B., Delbos, R.G., Banks, S.M.: The impact of overtime and long work hours on occupational injuries and illnesses: new evidence from the United States. Occupational and Environmental Medicine 62(9), 588–597 (2005)
6. Gorn, J.: The effects of music in advertising on choice behavior: A classical conditioning approach. The Journal of Marketing, 94–101 (1982)
7. Harrington, J.M.: Health effects of shift work and extended hours of work. Occupational and Environmental medicine 58(1), 68–72 (2001)
8. Hatem, P., Lira, I., Mattos, S.: The therapeutic effects of music in children following cardiac surgery. Jornal de Pediatria 82(3), 186–192 (2006)
9. Hu, X., Downie, J.S., Laurier, C., Bay, M., Ehmann, A.F.: The 2007 MIREX Audio Mood Classification Task: Lessons Learned. In: ISMIR, pp. 462–467 (September 2008)
10. Karageorghis, I., Terry, C.: The psychophysical effects of music in sport and exercise: a review. Journal of Sport Behavior 20(1), 54–68 (1997)
11. Pimenta, A., Carneiro, D., Novais, P., Neves, J.: Monitoring Mental Fatigue through the Analysis of Keyboard and Mouse Interaction Patterns. In: Pan, J.-S., Polycarpou, M.M., Woźniak, M., de Carvalho, A.C.P.L.F., Quintián, H., Corchado, E. (eds.) HAIS 2013. LNCS, vol. 8073, pp. 222–231. Springer, Heidelberg (2013)
12. Sparks, K., Cooper, C., Fried, Y., Shirom, A.: The effects of hours of work on health: a meta-analytic review. Journal of Occupational and Organizational Psychology 70(4), 391–408 (1997)
13. Yalch, R., Spangenberg, E.: Effects of store music on shopping behavior. Journal of Consumer Marketing 7(2), 55–63 (1990)

Detection of Distraction and Fatigue in Groups through the Analysis of Interaction Patterns with Computers

André Pimenta, Davide Carneiro, Paulo Novais, and José Neves

Abstract. Nowadays, our lifestyle can lead to a scatter of focus, especially when we attend to several tasks in parallel or have to filter the important information from all the remaining one. In the context of a computer this usually means interacting with several applications simultaneously. Over the day, this significant demand on our brain results in the emergence of fatigue, making an individual more prone to distractions. Good management of the working time and effort invested in each task, as well as the effect of breaks at work, can result in better performance and better mental health, delaying the effects of fatigue. This paper presents a non-intrusive and non-invasive method for measuring distraction and fatigue in an individual and in a group of people. The main aim is to allow team managers to better understand the state of their collaborators, thus preparing them to take better decisions concerning their management.

Keywords: Distraction, Fatigue, Task Performance, Behavioural Biometrics, Distributed Intelligence, Pattern Analysis.

1 Introduction

When people are working in a demanding cognitive task for an extended period of time, they often end up feeling the effects of fatigue, reflected in impaired task performance and reduced motivation to continue working on the task at hand [9, 10]. In addition, an increase in the amount and severity of errors being made can often be observed. Indeed, fatigue is considered one of the major causes for human failure and error [3]. An individual experiencing fatigue will also have a harder time

André Pimenta · Davide Carneiro · Paulo Novais · José Neves
CCTC/DI - Universidade do Minho Braga, Portugal
e-mail: {apimenta,dcarneiro,pjon,jneves}@di.uminho.pt

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_5

concentrating, getting easily distracted [2, 5], an indication that mental fatigue can have effects on selective attention.

Attention can be considered one of the most important mechanisms when it comes to acquiring new knowledge: being a cognitive process, attention is strongly connected with learning and assimilating new concepts either at school or at work [7]. The lack of attention can thus be problematic in some activities such as attending lectures, multimedia learning or car driving [1, 8]. Asides from learning, attention is also very important to perform any task in an efficient and adequate way and to distinguish between important and superfluous information for a given task at hand.

Our ability to focus on a task for prolonged periods is however at risk in a time in which our surroundings are constantly flooded with notifications, alerts and messages, coming from the operating system, friends, social networks and a myriad of (sometimes useless) applications, actively running or constantly in the background. Our brain, which did not evolve towards such a high-level of multitasking, may feel overwhelmed with the amount of sources information. This leads back to the importance of remaining focused on the task at hand, once that such ability is vital for any cognitive function, especially when there might be potential interference from distractors non pertinent to the task [6].

In this work it is detailed a monitoring system for distraction and fatigue based on the patterns of switching between leisure and work applications. Through the use of behavioural biometrics, specifically Keystroke Dynamics and Mouse Dynamics, we analyse the type of task performed by each user as well as the time spent in performing it. With this information we train classifiers that are able to distinguish scenarios in which the user shows sings of fatigue and distraction. This approach can be deemed both non-invasive and non-intrusive as it relies solely on the observation of the individual's use of the mouse and keyboard. This makes it more suited to be used continuously in work or academic environments than other available approaches as it requires no conscious or specific actions from the part of the user. Moreover, it is multi-modal as it relies on features that include the physical and behavioural modalities, potentially holding better results than single-modality approaches.

2 A Non-intrusive Approach to Monitoring Distraction and Fatigue

This paper introduces an approach for determining the type of task being performed on a computer by an individual or by a group. Tasks such as reading, writing reports or programming are examples of tasks that require a significant amount of attention and can be performed using a wide range of different tools. However, what all these tools have in common, is that they are interacted with using mouse and keyboard. The use of these peripherals, as addressed in [4], allows to acquire contextual features that describe the interaction patterns of the user with the computer. These features reflect the behaviour of the user and how it changes under certain

conditions, such as when the user is fatigued. In his particular case, we look at how these features change when the user starts becoming distracted.

Particularly, we look at how the user distributes the time devoted to each application, and at which of these applications are related with the task at hand. We thus complement and improve the previously developed fatigue monitoring approach (task-independent, based solely on the performance of the interaction with the computer) with a measure of attention, which can be used as a reliable indicator of fatigue [7].

2.1 Methodology

In order to analyse the proposed problem, a study was set up aimed at collecting the necessary data. The methodology followed to implement the study was devised to be as minimally intrusive as the approach it aims to support. Twenty seven (20 men, 7 women) participants, students from the University of Minho, were selected to participate. The participants were provided with an application for logging the system events related to the mouse and keyboard as described in [4].

It is based on these events that we build the features that described the interaction of each individual over time. Moreover, the provided application also logs the application being used at any moment. This application, which maintained the confidentiality of of the users and of their interactions, needed only to be installed in the participant's computer and would run on the background, starting automatically on system start. The only explicit interaction needed from the part of the user was the input of very basic information on the first run, including the identification and age.

This application was kept running for approximately one month, collecting interaction data whenever there was interaction with the computer. During the whole process, participants did not need to perform any additional task and were instructed to perform their activities as usual, whether they were work or leisure-related, as they would if they were not participating on the study.

When the period of one month ended, participants were asked to send in the file containing the log of their interaction during the duration of the study. The resulting dataset as well as the process by means of which data were analysed is described in the following sub-sections.

The events generated by the mouse and keyboard were divided into 4 categories: (1) *Chat*, concentrating inputs in applications such as Skype, Hangouts, Facebook Mensseger; (2) *Browsing*, including inputs in browsers such as Internet Explorer, Firefox and Google Chrome; (3) *Work* , with inputs regarding applications such as as Eclipse IDE, Microsoft Office Suite, TexMaker, Adobe Reader, Evernote or Netbeans; and (4) *Games*, with records of gaming applications.

2.2 Analysis of Results

In this section we show the existence of different behaviours in using the keyboard and mouse according to the type of task being performed. We do so, as previously discussed, through the analysis of the interaction patterns of the individuals. Specifically, we look at the distributions of the data collected for each category of application and analyse the statistical significance of their differences. To this aim, the following approach was implemented.

First, it was determined, using the Pearson's chi-squared test, that most of the distributions of the data collected are not normal. Given this, the Kruskal-Wallis test was used in the subsequent analysis. This test is a non-parametric statistical method for testing whether samples originate from the same distribution. It is used for comparing more than two samples that are independent, and thus prove the existence of distinct behaviours. The null hypothesis considered is: H_0 : all samples have identical distribution functions against the alternative hypothesis that at least two samples have different distribution functions.

For each set of samples compared, the test returns a *p*-value, with a small *p*-value suggesting that it is unlikely that H_0 is true. Thus, for every Kruskal-Wallis test whose $p - value < \alpha$, the difference is considered to be statistically significant, i.e., H_0 is rejected. In this work a value of $\alpha = 0.05$ is considered, a standard value generally accepted by research.

This statistical test is performed for each of the features considered, with the intention of determining if there are statistically significant differences between the several distributions of data, which will in turn confirm the existence of different behaviours in using the keyboard and mouse on the type of task being performed through interaction patterns.

For 90% of the users all the features showed statistically significant differences. Figure 1 depicts these differences clearly for two different features: writing velocity and mouse velocity. It is possible to conclude, for example, that participants write the fastest when in chat applications and move their mouses faster when playing games.

3 Distributed Intelligence Architecture

As can be seen in Section 2 the use of keyboard and mouse reveals distinct behaviours when participants are working or when they are distracted. Based on the results briefly described it was developed a prototype of an application that aims to classify the attention of the user according to their interaction patterns.

The proposed framework aims to assess the level of attention in scenarios of work or study of individuals. Nonetheless, the main objective is to support the decision-making processes of team managers or group coordinators. In this perspective, each element of a group is seen as part of a whole which contributes to the general level of fatigue and the distraction of the group. We will call each member of the group a

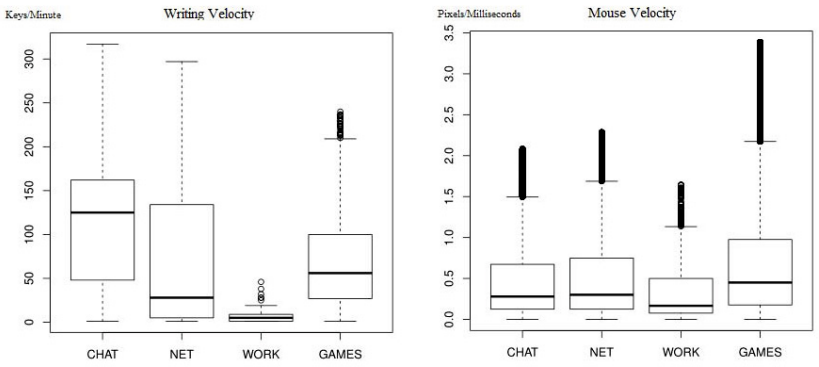


Fig. 1 Differences in the use of mouse and keyboard between the different categories of applications

"Monitored User", with the exception of the coordinator/manager who will is called "Coordinator". Figure 2 depicts the distribution of the roles and the flow of information from raw data to the classification of the state of the individuals.

Each monitored user provides raw data about their interaction patterns to the coordinator. The information provided results of registering the events related to keyboard and mouse. The data collected is processed and transformed in order to be evaluated in terms of the features mentioned. One of the most important tasks is to filter outlier values that would have an undesirable effect on the analysis (e.g. when we continuously press the backspace key to delete a group of characters).

After the data has been processed it is classified and it is used to build the metadata that will support decision-making. To do it, the machine learning mechanisms detailed below are used. At this stage the process of classification can be improved with the inclusion of information from work settings and other scenarios. The information compiled can be presented to the respective user and to the coordinator, for improving routines and decision-making.

The computer of the coordinator receives this information in real-time and calculates, at regular intervals, an estimation of the general level of fatigue and attention of the group. The coordinator has access to the current and historical state of the group from a global perspective, but can also refer to each user individually (Figure 4).

The process described can be found in Figure 3, which details the process of monitoring.

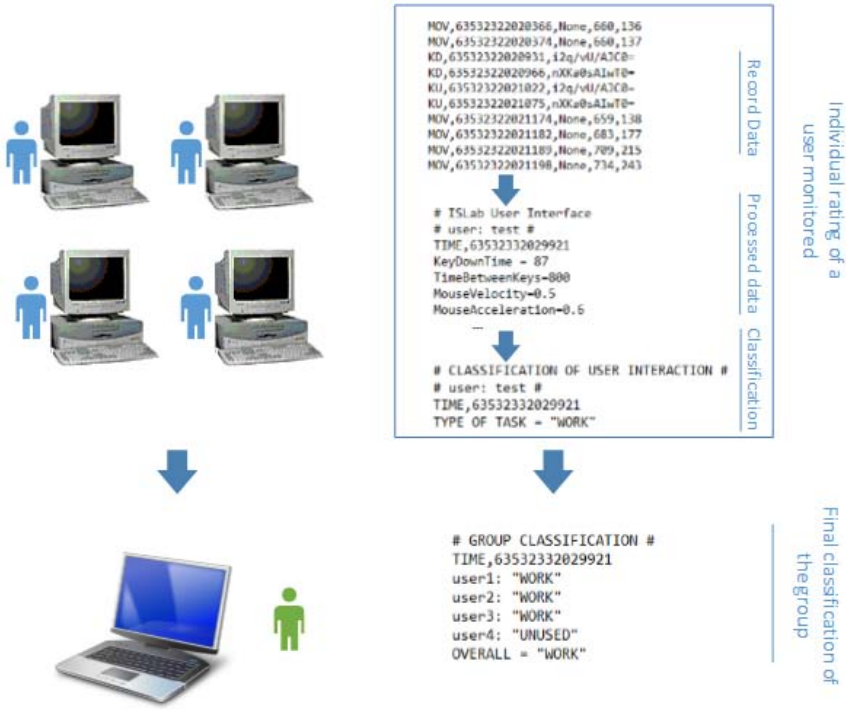


Fig. 2 Flow of information from raw data to the classification of the state of the individuals. The coordinator is depicted in green while users are depicted in blue.

3.1 Classifying Behaviours Associated with Work Tasks

Having proved the existence of distinctive behaviours in the use of the keyboard and mouse in tasks of different types, a classifier was built and trained to determine the type of task being carried out by the users from their interaction patterns.

The classification of the type of task being performed by the user is obtained through use of (k-NN) k-Nearest Neighbor algorithm. It is a classification method based on closest training examples in the feature space. The data used to train the model used by the k-NN algorithm were collected in an experiment described in Section 2. The classification of the state of the group is the result of the average of all monitored users.

The following section describes a case study in which the classifier trained was tested and validated with different participants, in the context of a classroom.

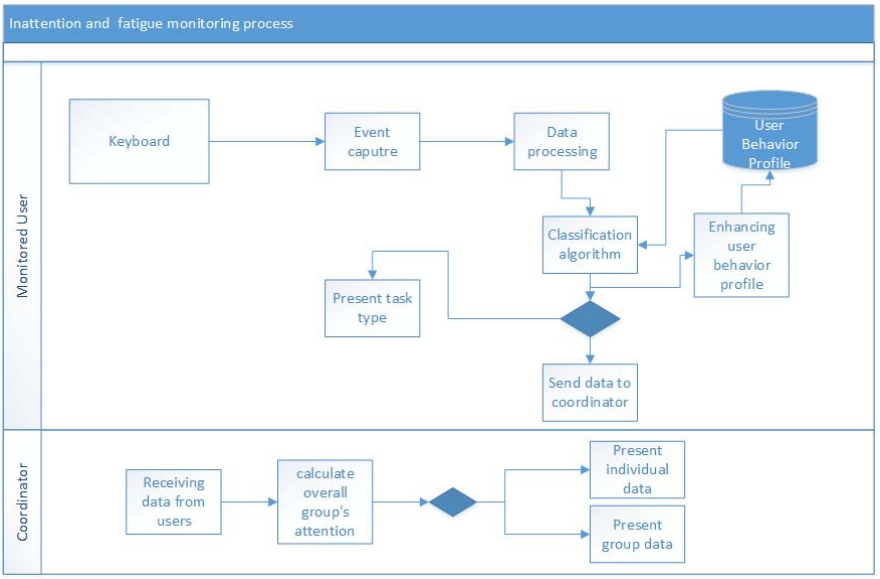


Fig. 3 Work-flow of information in the monitoring process

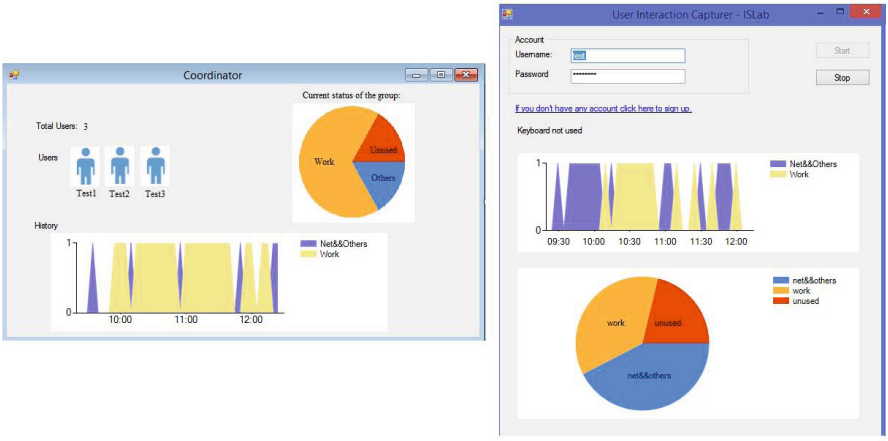


Fig. 4 The presentation layer of the monitoring system detailing the state of the group (right), and the individual details of a user (left)

4 Case Study

In order to test the developed system and approach a case study was set up, including the roles of users and coordinator. We meant to analyse to which extent the coordinator could find the information provided interesting.

In addition to testing and validating the system we were also looking for effects of distraction and fatigue. In that sense we searched for potential activities that needed a significant amount of attention for its correct execution.

A training course on programming was chosen for testing this approach in a real scenario. The course takes place physically in a classroom and comprises a coordinator who is responsible for teaching a programming language (in this case C) to a group of participants. The participants in the case study, eighteen in total (13 men, 5 women) were students from the University of Minho of the field of physical sciences. Their age ranged between 18 and 25.

Each session of the training course has a duration of 3 hours, which always follow the same "protocol": some theoretical concepts are introduced at the beginning of the class and the rest of the session is spent practising and solving exercises using the computer and a specific IDE.

At the end of each session a practical evaluation exercise is carried out for each individual, aimed to determine how well the concepts thought were perceived. The computer is accessible and the students may use it as they wish during the duration of the session, both while the coordinator provides the theoretical background and while students use the IDE. Participants frequently use the computer to take notes, search additional information, send emails or visit social networking sites. There is no restriction on the use of the computer expect for its mandatory use for solving exercises.

4.1 Results

During a training session, the developed monitoring system was used to assess the level of attention of the participants, as well as its performance. Given the domain of the case-study, the participant's usage of the peripherals was classified as belonging to a work-related application or to any other one. At the end of the class, participants were evaluated with specific exercises. Our aim is to find a relationship between distraction, fatigue, interaction performance and scholar performance. Participants were also requested to rate their level of attention during the session using a value between 0 (very distracted) and 5 (attentive), in order to validate the results from a subjective point of view.

The results achieved are available in Table 2. It can be observed that participants with the lowest attention during the session had worse evaluations unlike participants who were more attentive and obtained better results. The participants number 4, 6, 8, 14, 15, 17 are examples of participants who did not pay much attention during the session and had worse results. These same participants also confirmed, through their answers in the questionnaires, that they were not very focused on the class.

Considering the correlation between variables, the following positive correlations were identified: correlation between the percentage of time spent interacting with work-related applications and final score (0.54), correlation between the percentage

Table 1 Overall results of all participants, where one can see the values of the monitoring, evaluation and values of the subjective level of attention to each participant

user	Minutes in work tasks	Minutes others tasks	Minutes unused	% work	% others	% subjective attention	% final evaluation
1	96	36	48	73%	27%	5	90%
2	60	36	84	63%	38%	4	100%
3	72	12	96	86%	14%	4	100%
4	54	48	78	53%	47%	2	50%
5	90	0	90	100%	0%	3	50%
6	54	90	36	38%	62%	3	30%
7	90	12	78	88%	12%	4	100%
8	60	42	78	59%	41%	4	70%
9	96	42	42	70%	30%	4	100%
10	108	0	72	100%	0%	5	80%
11	84	24	72	78%	22%	4	90%
12	96	24	60	80%	20%	5	100%
13	84	24	72	78%	22%	3	70%
14	48	42	90	53%	47%	3	50%
15	54	36	90	60%	40%	3	0%
16	60	30	90	67%	33%	4	90%
17	48	54	78	47%	53%	3	20%
18	114	30	36	79%	21%	5	80%
Mean	76	32	71	71%	29%	4	70%

of time spent interacting with work-related applications and subjective evaluation of attention (0.51) and correlation between subjective evaluation of attention and final score (0.68).

4.2 Validation

In order to validate the results obtained by the monitoring system during the experiment, the applications used by the participants were recorded, as well as the time spent in each one.

The only application that had to be mandatorily used was the Dev-C++ IDE. Nonetheless, some participants used alternative text editors such as Microsoft Office, TextMaker or Evernote. Thus, we looked at what type of application was being used in each moment and what type of application the classifier was providing as output to validate its efficacy.

The results obtained during the scan for all classifications from all users during the formation validate the classifications made by the monitoring system. As depicted in Table 2, 96% of classifications as work tasks through the use of work tools were successful, while the remaining 92% of applications had successful classifications as well. It was also observed that the keyboard was used for tasks other than work-related ones, mostly chat applications and browsers.

Table 2 Results of the classifications algorithm

task type	instances	% correctly class.	% incorrectly class.
work tasks	228	96%	4%
others	97	92%	8%
Unused	215	100%	0%

5 Conclusion

This paper described a prototype for monitoring the type of task being carried out by an individual or a group of individuals. The main aim is to detect patterns associated to distraction and fatigue in scenarios of classrooms or workplaces. The information compiled is provided to team managers and coordinators so that they can improve their decision-making skills. In the example of the case-study, the coordinator of the class gains a better notion of how long he can talk before starting to loose the attention of the class. Students, on the other hand, learnt that there is a direct relationship between the attention they devote to the contents of the class and their final score.

The results also show that carrying our different types of task results in entirely different interaction patterns. These interaction patterns are different enough to train classifiers that are able to classify the type of task being carried out. This can then be used to determine the distribution of time of the individual among the several types of applications. Within the context of CAMCoF project, which founds this work, the long-term goal is to develop environments that are autonomous and take measures concerning their self-management to minimize fatigue and increase the performance and well-being of a group of individuals.

Acknowledgements. This work is part-funded by ERDF - European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project FCOMP-01-0124-FEDER-028980 (PTDC/EEI-SII/1386/2012).

References

1. Alm, H., Nilsson, L.: The effects of a mobile telephone task on driver behaviour in a car following situation. Accident Analysis & Prevention 27(5), 707–715 (1995)
2. Bartlett, F.C.: Ferrier lecture: fatigue following highly skilled work. Proceedings of the Royal Society of London. Series B-Biological Sciences 131(864), 247–257 (1943)
3. Boksem, M.A., Meijman, T.F., Lorist, M.M.: Effects of mental fatigue on attention: an erp study. Cognitive Brain Research 25(1), 107–116 (2005)
4. Carneiro, D., Novais, P., Catalão, F., Marques, J., Pimenta, A., Neves, J.: Dynamically improving collective environments through mood induction procedures. In: van Berlo, A., Hallenborg, K., Rodríguez, J.M.C., Tapia, D.I., Novais, P. (eds.) Ambient Intelligence - Software & Applications. AISC, vol. 219, pp. 33–40. Springer, Heidelberg (2013)

5. Faber, L.G., Maurits, N.M., Lorist, M.M.: Mental fatigue affects visual selective attention. PloS One 7(10), e48073 (2012)
6. Horvitz, E., Jacobs, A., Hovel, D.: Attention-sensitive alerting. In: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, pp. 305–313. Morgan Kaufmann Publishers Inc. (1999)
7. Hwang, K., Yang, C.: Automated Inattention and Fatigue Detection System in Distance Education for Elementary School Students. Journal of Educational Technology & Society 12, 22–35 (2009)
8. Mayer, R.E., Moreno, R.: A split-attention effect in multimedia learning: Evidence for dual processing systems in working memory. Journal of Educational Psychology 90(2), 312 (1998)
9. Meijman, T.F.: Mental fatigue and the efficiency of information processing in relation to work times. International Journal of Industrial Ergonomics 20(1), 31–38 (1997)
10. van der Linden, D., Frese, M., Meijman, T.F.: Mental fatigue and the control of cognitive processes: effects on perseveration and planning. Acta Psychologica 113(1), 45–65 (2003)

Lifestyle Recommendation System for Treating Malnutrition

Cristina Bianca Pop, Viorica R. Chifu, Ioan Salomie,
Adela Stetco, and Roxana Plaian

Abstract. This paper presents a system for treating malnutrition by generating dietary recommendations according to the person's health profile. The system consists of monitoring, analyzing and recommending components. The monitoring component collects information about a person's nutrition habits. The analyzing component classifies the nutrition habits as having a risk of developing malnutrition or not. If an unhealthy nutrition habit is detected, the recommending component generates appropriate dietary recommendations using a Honey Bees Mating Optimization based algorithm.

1 Introduction and Related Work

Malnutrition refers to an unhealthy condition trigerred by an unbalanced diet featured by a deficit or excess of one or more nutrients that are required for a healthy life. Current statistics state that 15 to 50 % of the elder population is affected by poor nutrition and malnutrition caused by several medical, lifestyle, social, and psychological factors [2]. Poor nutrition/malnutrition may contribute to or exacerbate chronic/acute diseases,or may speed up the development of degenerative diseases.In this context, various systems for generating personalized food recommendations have been proposed in the research literature. For example, in [3] the authors present a property-based collaborative filtering strategy which recommends specific products or customized diets for a person based on his electronic health records. The proposed strategy consists of building a matrix of values, each value specifying how suitable is a property of an item for a user property. In [4], the authors propose a system for recommending restaurants that provide food menus for diabetes.

Cristina Bianca Pop · Viorica R. Chifu · Ioan Salomie · Adela Stetco · Roxana Plaian
Tehnical University of Cluj-Napoca, Computer Science Department,
Baritiu 26-28, Cluj-Napoca, Romania
e-mail: {Cristina.Pop,Viorica.Chifu,Ioan.Salomie}@cs.utcluj.ro

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_6

A collaborative filtering based prediction method has been integrated within the system to filter and rank the restaurants based on the user preferences.

In this paper we present a system for treating malnutrition by generating dietary recommendations according to the person's health profile. The system consists of monitoring, analyzing and recommending components. The monitoring component collects information about a person's nutrition habits. The analyzing component classifies the person, based on the nutrition habits, as having malnutrition or not by means of decision trees. If malnutrition is detected, the recommending component generates appropriate dietary recommendations using an HBMO based algorithm [1].The paper is structured as follows. Section 2 briefly describes the conceptual architecture of the Lifestyle Recommendation System, while Section 3 presents the Lifestyle Analysis and Recommending modules. In Section 4, a case study illustrating the system functionality is presented. The paper ends with conclusions.

2 System Conceptual Architecture

The conceptual architecture (see Figure 1) of the proposed prototype is composed of the Lifestyle Profile Generator module, the Lifestyle Analysis module, and the Recommending module whose functionalities are exposed as Web services. The Lifestyle Profile Generator module creates the lifestyle profile of a person based on the information collected from (1) wireless sensor networks, motion detection devices, and other sensors incorporated in the home of a person, and (2) health monitoring devices attached to the person's body. The lifestyle profile is further integrated in the personal profile of the monitored person that additionally contains information regarding the person's medical history collected from medical records, and nutrition and physical preferences collected by means of questionnaires. The personal profile of a person is used by the Lifestyle Analysis module to detect unhealthy behaviors

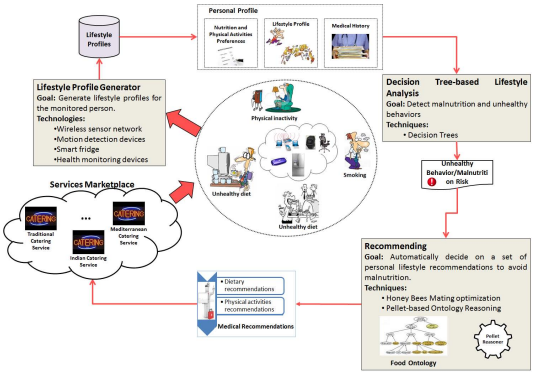


Fig. 1 The Architecture of the Lifestyle Recommendation System

or malnutrition risks by means of decision trees. In case an unhealthy behavior or a malnutrition risk is detected, the Recommending module is triggered to generate dietary recommendations using a HBMO-based algorithm, the Pellet reasoner [17], and a Food Ontology built based on the information provided in [11].

3 Healthy Lifestyle Recommendation

This section presents the two steps used for generating healthy lifestyle recommendation.

Step 1: Decision Tree-Based Lifestyle Analysis. To establish whether a monitored person suffers of malnutrition, a classification of the person's personal profile is done. We have used a decision tree-based algorithm [9] to classify the personal profile. A training set with the following types of attributes has been defined for training the decision tree-based algorithm: % of weight loss, no nutritional intake for more than five days, oedema presence, current body mass index, and mid-upper arm circumference. The training set's records are assigned to one of the following class labels: malnutrition presence, or malnutrition absence.

Step 2: Healthy Lifestyle Recommendation. The process of generating healthy lifestyle recommendations consists of two stages: identifying the optimal nutritional features, and generating the optimal nutritional diet.

Stage 2.1: Identifying the Optimal Nutritional Features. This stage identifies the optimal quantity of carbohydrates, proteins, fats and calories that should be consumed by a person to avoid malnutrition.

Step 2.1.1: Compute the current BMI of a person as in [12].

Step 2.1.2: Compute the kilograms number that must be gained/lost as:

$$kilos_{gain/lost} = (BMI_c - BMI_n) * height^2 \tag{1}$$

where BMI_c is the current BMI, BMI_n is a BMI value in the normal range [18.5, 24.9], and $height$ is the person's height in meters.

Step 2.1.3: Compute the number of calories per day. To gain or lose the number of kilograms computed in Step 2.1.2, the number of calories that should be consumed daily must be computed by taking into account the fact that it is healthy to lose/gain at most 5%-10% of the current weight monthly. The formula to compute the number of calories is defined as [6]:

$$cal_{req} = \frac{3500 * \frac{(weight_c - \frac{w}{100} * weight_c) * 0.453592}{4}}{7} + cal_{daily} \tag{2}$$

where $weight_c$ is the current weight of the person, w is a value in [5,10], and cal_{daily} is the current average number of calories that the person consumes. This formula is applied to compute the number of required calories that must be consumed each month until the desired weight is achieved.

Step 2.1.4: Compute the number of carbohydrates per day. As one gram of carbohydrates corresponds to 4 calories [7] and the quantity of carbohydrates consumed daily should represent 55% of the total daily calories [8], we use the following formula to compute the daily number of carbohydrates:

$$carbs_{req} = \frac{55\% * cal_{req}}{4} \tag{3}$$

Stage 2.2: Generating the Optimal Nutritional Diet. The algorithm (Algorithm 1) for generating the optimal nutritional diet is based on the HBMO algorithm [1] inspired by the mating behavior of queen bees in nature.In the HBMO algorithm, the queen and the drones it mates are solutions of the optimization problem being solved, while the workers are heuristic strategies used to update solutions. Just as in HBMO, in the process of generating the optimal nutritional diet we have a queen agent and a set of drone agents which have a nutritional diet solution associated. A nutritional diet is represented as a set of food items consumed at breakfast, at snacks, at lunch, and at dinner and is evaluated based on the total number of calories and carbohydrates of the food items part of the diet. A solution is considered as optimal or near-optimal if the total number of calories and carbohydrates is equal or close to the optimal nutritional values. The HBMO-based algorithm takes as input the optimal nutritional features computed in the previous stage, and the percentage of the best diets that are kept from one iteration to another.The algorithm steps for generating a diet recommendation for one day are organized in an initialization stage, and an iterative stage. In the initialization stage, a set of diets is randomly generated and the best one is identified. In the iterative stage, the following steps are performed until a diet having the total number of calories and carbohydrates equal or close (+/-20% of the total values) to the optimal nutritional values is identified: (i) *Update diets*. In this step, a crossover is applied between the best diet and the other generated diets.As a result two new diets are obtained, and the one having the lowest fitness is submitted to a mutation process. (ii) *Update best diet*. After the crossover and mutation processes are completed, the best diet is updated. (iii) *Update the set of diets*. A new set of diets composed of a percentage of the current set of the diets having the highest fitness and a set of randomly generated diets will be submitted to the next iteration.

4 Case Study

The proposed HBMO-based method has been tested on users suffering of diabetes. In this section we consider a case study for a person that has been monitored for 30 days and has the personal profile presented in Figure 2.

Table 1 presents the monitored person's behaviour summary for the 30 days. The behavioral summary contains information about: the average number of calories consumed per day, the average quantity of carbohydrates consumed per day, the % of the weight loss/gain and kilograms compared to the last month, the old and new value of the weight, the old and new value of the BMI, the old and new value of the

Algorithm 1. HBMO-based_Recommendations_Algorithm

Input: $nFeat_{opt}$ - optimal nutritional features, p - the percentage of solutions kept from one iteration to another
Output: $Diets_{opt}$ - optimal or near-optimal diets for a week
begin
  $Diets_{opt} = \emptyset$
  for $i = 1$ **to** 7 **do**
    $Diets =$ **Generate_Random_Diets**()
    $diet_{best} =$ **Select_Best_Diet**($Diets$)
    while (!**Stopping_Condition**($nFeat_{opt}$)) **do**
      $Diets =$ **Crossover**($diet_{best}, Diets$)
      $Diets =$ **Mutation**($BroodDiets, Diets_{opt}$)
      $diet_{best} =$ **Select_Best_Diet**($BroodDiets$)
      $RandomDiets =$ **Generate_Random_Diets**($100\% - p$)
      $Diets =$ **Replace_Worst_Diets**($RandomDiets$)
    end while
    $Diets_{opt} = Diets_{opt} \bigcup \{diet_{best}\}$
  end for
  return $Diets_{opt}$
end

Height	*Weight*	*MUAC*	*Oedema*	*Physical Activity*	*Sleep time*	*Stress level*	*Alcohol*	*Cigarettes*	*No of Meals*
1.70 m	70 kg	21.0 cm	No	30 minutes/day	6-9 hours/night	medium	2 units/day	5 cigarettes/day	2

Breakfast favorite items	*Lunch favorite items*	*Dinner favorite*	*Snacks*	*No of lunch dishes*
Omelet, Cheese	Roasted chicken, Chicken noodle soup	Caesar Salad	Banana	1

Fig. 2 Personal Profile of a person

MUAC, and the malnutrition status. Because the malnutrition has been detected, a set of recommendations are given to improve the health status of the person and to eliminate malnutrition. First, the following optimal nutritional features are identified such that the BMI of the monitored person reaches 23.01 in a period of 3 months: the recommended number of calories that should be consumed per day is 2356 kcal out of which 55% should be carbohydrates, 15% should be proteins, and 30% should be fats. Then, based on these facts, the HBMO-based algorithm generates lifestyle recommendations regarding nutrition for a period of seven days. In Figure 3 we illustrate diet recommendations generated for two days.

Table 1 Behaviour summary for 30 days

Calories	*Carbo*	*Weight loss/gain*	*Loss/gain*	*oWeight*	*nWeight*	*oBMI*	*nBMI*	*oMUAC*	*nMUAC*	*Malnutrition*
1106kcal/day	124.77 g/day	-7.84%	-5.49kg	70.00kg	64.51kg	24.22	22.32	21.00cm	20.45cm	present

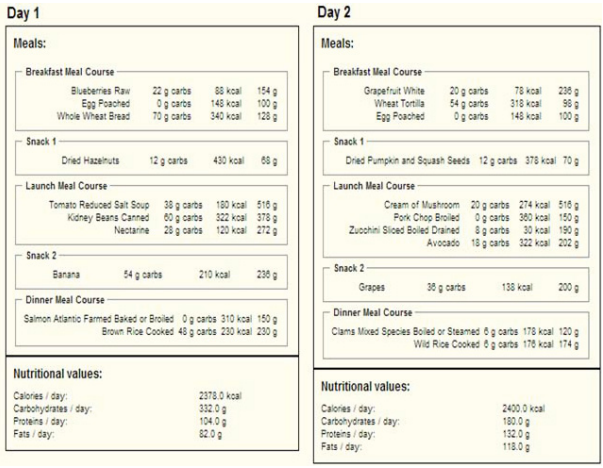


Fig. 3 Example of diet recommendations generated for two days

5 Conclusion

This paper has presented a lifestyle recommendation system for avoiding and treating malnutrition. The system integrates a Decision Tree-based classifier to determine whether a person suffers from malnutrition, and a HBMO-based algorithm for generating the optimal lifestyle recommendations to improve the health status of a person thus ensuring that malnutrition is treated.

Acknowledgements. This work is carried out under the AAL Joint Programme with funding by the European Union (project AAL-2012-5-195) and is suported by the Romanian National Authority for Scientific Research, ï£¡ UEFISCDI, (project AAL - 16/2013).

References

1. Haddad, O.B., et al.: Honey-Bees Mating Optimization Algorithm: A New Heuristic Approach for Water Resources Optimization. Water Resources Management Journal 20(5), 661–680 (2006)
2. Hickson, M.: Malnutrition and Ageing. Postgraduate Medical Journal 82(963) (2006)
3. Lopez-Nores, M., et al.: Property-based Collaborative Filtering for Health-aware Recommender Systems. In: Conf. on Consumer Electronics, pp. 345–346 (2011)
4. Chin, C.M.: Mobile Health Monitoring: The Glucose Intelligence Solution. In: TR (2012)
5. Suksom, N., et al.: A Knowledge-based Framework for Development of Personalized Food Recommender System. In: ICKICSS (2010)
6. http://www.livestrong.com/article/296908-how-many-calories-does-it-take-to-burn-1-pound-of-fat/
7. http://www.nutristrategy.com/nutrition/calories.htm
8. http://www.mensfitness.com/nutrition/what-to-eat/the-fit-5-using-carbs-wisely
9. Weka, http://www.cs.waikato.ac.nz/ml/weka/
10. Pellet Reasoner, http://clarkparsia.com/pellet/
11. http://ndb.nal.usda.gov/ndb/search/list
12. http://www.cdc.gov/healthyweight/assessing/bmi/adult_bmi/

Part III
Agents

Distributed Event Processing for Goal-Oriented Workflows

Kai Jander, Lars Braubach, and W. Lamersdorf

Abstract. Goal-oriented workflows enable workflow designers to easily include a high degree of flexibility while implementing and automating business process. In addition, modeling is based on the business goals of the organization instead of actions, allowing for better alignment with those goals. Combined with a distributed workflow management system, this allows for a highly agile workflow environment that can adapt to the business situation while maintaining the structure of a solid workflow model. However, one of the drawback of such processes is a lack of attribution of the actions of the workflow to specific goals. As a result, improvements to the monitoring side of process management are necessary in order to make such associations clearer and allow easier workflow analysis and reengineering. This paper presents an approach for a component-based event system that introduces a degree of structure to the events, enabling the association of events with the workflow model, facilitating real-time monitoring of goal-oriented process and process drill-down analysis.

1 Introduction and Motivation

In business process management, workflows represent the partially automatized part of a business process implemented to be executed on a computer system. While they play an important part of business automation, traditional workflow models such as the Business Process Model and Notation (BPMN) [13] often lack good support for flexibility in a multitude of business situations, requiring the modeler to include numerous branches for every conceivable situation. This deficit has lead to a number of different approaches like ADEPT [12] attempting to attenuate those limitations. Part of the difficulties stem from the fact that most approaches to workflows are

Kai Jander · Lars Braubach · W. Lamersdorf
Distributed Systems and Information Systems Group, University of Hamburg, Germany
e-mail: {jander,braubach}@informatik.uni-hamburg.de

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_7

activity-based, which means they are focused on a specific order of actions with branching explicitely inserted at certain points. While this is a fairly intuitive approach for modeling workflows, it does not directly provide business reasons for activities, which can result in the inclusion of unnecessary or unwanted activities that merely exist for technical rather than business reasons in the workflow.

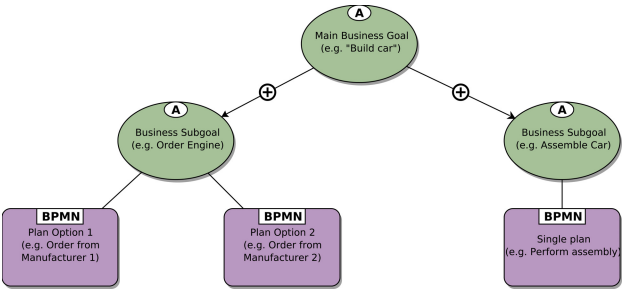


Fig. 1 Example of a simplified goal-oriented process

An interesting concept for addressing both concerns are goal-oriented workflows [10]. In this approach, Belief-Desire-Intention (BDI) agents [4] are used to model workflows. The BDI approach allows the workflows to be based on business goals (see Figure 1), starting with the overall goal the business aims to accomplish with the workflow which is then subdivided into multiple subgoals with various types of interactions such as priorities and inhibitions. These subgoals taken together represent the top-level goal by completely covering all aspects of that goal. The subgoals themselves can again be further decomposed until the goals are plain enough to be achievable by a simple action-based workflow. If multiple solutions for a goal are possible, multiple plans can be attached to the goal and a plan is chosen at runtime based on attached conditions and attributes. Key for both the conditions and attributes of goals and plans is the existence of a workflow context, which not only holds the internal state of the workflow but also contains information about the current business situation, allowing this information to be accessed by goal and plan conditions.

The methodology of generating GPMN models is similar to Hierarchical task network (HTN) [5] planning approaches, however, while the goal-structure of GPMN processes often end up being hierarchical, they do have to be. In addition, GPMN-modelling is not used for feasibility analysis or ahead-of-time planing. Rather, it allows process engineers to specify business contingencies and competing objectives for long-running processes in dynamic environments such as R&D processes at Daimler AG [9]. Instead, the resulting agent model can be directly executed by a BDI interpreter, which uses a BPMN interpreter for concrete plans. The reasoning of BDI is split between the goal deliberation cycle, which decides which of potentially several conflicting goals to follow and the means-end reasoning which choses pre-defined plans to achieve selected goals. The BDI interpreter reasoning engine uses an approach called Easy Deliberation [16] to resolve this cycle.

While this top-down and business-driven approach towards workflow modeling is very intuitive and flexible, its adaptive behavior during runtime means that actions are not always attributable to the goals [8]. In addition, goal-based workflows are also often used in dynamic environments where not only the workflows but also the workflow management system are distributed to increase flexibility and robustness [11]. As a result, a distributed monitoring system is necessary that links the actions performed in the workflow with the goals and plans representing the business reasons.

Based on the requirements for a distributed workflow management system, distributed workflow execution and cohesion between goals and action, the proposed system should be able to meet the following objectives:

- Goal-Action Cohesion
  - The primary goal of the approach is providing event information that allow attribution of concrete business actions with business goals in a workflow.
  - The user must be able to perform a "drill-down" analysis to trace causes of events from the most detailed to the most high-level goals.
- Distributed Workflow Management
  - The system must be adaptable to a wide variety of business infrastructures, including transient and mobile systems and must respond in a robust fashion to changes in that infrastructure.
  - As a result, the system must be distributed, redundant, robust and adaptable. Communication must be kept low for efficiency.

In this paper we present a component- and service-based approach attempting to solve these objectives. It supports monitoring distributed goal-based workflows using a hierarchical structure of events combined with a distributable monitoring system for gathering and redestribution of the events to the workflow clients.

2 Related Work

The approach presented in this paper primarily touches two important area of research. The first is the area of *Business Activity Monitoring (BAM)*. This area concerns itself how business transactions happening within a company can be recorded and processed, either retroactively or in realtime. The most common approach to this challenge is to record business transactions, either specifically for monitoring purposes or incidentally as part of regular business record keeping, in a *data warehouse* [4]. In addition, *extract, transform, load (ETL)* processes can be used to extract additional data from other sources within the business [18]. The resulting data can be utilized in two ways: Complex analysis can be performed and long-term statistics can be gained by applying data mining techniques to the data available in the data warehouse [1]. This offers the user an in-depth and long-term perspective

of the perfomance of the organization. However, it can require substantial time and computation to process the available data and thus may lag behind the current development of the business. For example, *online analytical processing (OLAP)* allows for multidimensional analysis of transactions and currently available data within the organization [2], but the data must already be available in a structured fashion, for example in a data warehouse. The data is processed to allow the user to approach it from multiple perspectives using multidimensional analysis [17] by forming structures such as OLAP cubes.

Alternatively, realtime monitoring can be achieved using *complex event processing (CEP)* [6]. This approach gathers and processes events as a stream and generates useful information for the BAM system. While aspects of this approach are similar to the approach presented here, it is focused on the processing of data, rather than providing a stronger coherence between the operational and strategic level by aligning actions and goals.

The resulting information can then be used to display information in a dashboard specific to the interest to the business user. Statistics and indicators are provided to allow the user to monitor the actual performance of the business and compare it with previously strategically defined *key performance indicators (KPI)*, which represent quantifiable values. However, the relationship between the processed data and the KPIs is implicit and cannot be derived from the data warehouse itself. As a result, the dashboard functionality of BAM systems often need to be customized to restore this relationship, which can only be partially compensated through standardization of common KPIs or interpretation of standardized charts on the dashboard. Furthermore, BAM focuses primarily on statistical data and thus does not provide an easy way for attributing transactions of the business with particular strategic goals. In contrast, the approach presented here focuses on attributing actions and tasks to business goals, while the generation of statistics is less of a concern. In addition, data mining solutions often focus on a centralized data warehouse solution while the monitoring system presented here provides distributed realtime monitoring.

The second area concerns itself with distributed event systems [14]. A common approach here is to employ *event brokers*, which receive published events from *event publishers* and then distribute them among the other brokers within the system, making them available to *event subscribers* throughout the system. Similar to this approach, we use one or more monitoring component instance to implement the broker functionality and distribute our goal-based events as part of a larger distributed workflow management system. The number of such brokers used can be chosen by the user of the system, with additional brokers increasing the redundancy and thus the resilience of the overall system and increasing the event processing performance by bundling event transfers.

3 Event Structure

The targeted goal-oriented workflows are based on the Jadex Active Components [19] approach. This approach offers a number of different active component types such as micro agents, BDI agents, BPMN workflows and the goal-oriented GPMN workflows, which are modeled as goal-plan hierarchies but are translated to BDI agents for execution.

Events generally denote an occurence associated with an *event source*. An event source can be any entity or part of an entity within the distributed system such as an active component itself or some part of the component. In order to broadly distinguish events based on sources such as components, goals and plans, a event source can be attached to the event. A number of default categories are already provided for all active component types, including the component category for events emitted by component instances itself, the execution category for the execution of components steps, the service category for component services and the property category for component property. More specialized categories are available for action-oriented and goal-oriented workflows. Event source categories for activities, goals and plans are available in addition to the workflow context fact category. The latter category is also an example of an event source category where the modification event applies.

While the exact nature of the event can vary wildly, a number of categories can be identified that not only define the relationship with the event source and between events over time. For example, BPMN offers three broad classes of event types: Start events, which mark the start of a process, end events, which denote the process end and finally intermediate events which can occur while the process is running.

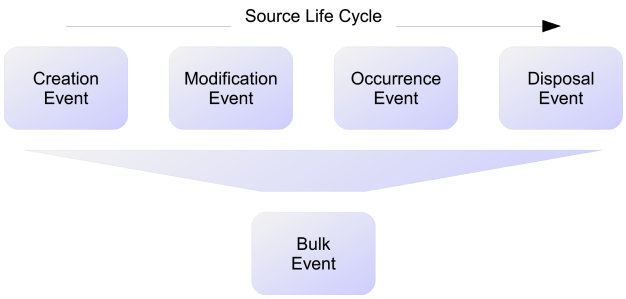


Fig. 2 Event types used over the event source life cycle

Similar to this, we have based our system on five different types of events based on the life cycle of the source (see Figure 2). The first type are *creation events*, which are issued when the event source is first created, providing similar semantics as BPMN start events. The counterpart for end events are *disposal events*, issued when the life cycle of the event source has ended and the source has been disposed. During the life cycle, two additional types of events can occur: When the state of the source changes it results in a *modification event*, while the other event type is the

occurrence event, which simply denotes a point in time when an action took place. For example, an occurrence event is generated when an agent receives an external message or an internal user event is triggered. Finally, a special type of event, the bulk event, only applies when multiple events are aggregated into a single event for efficiency when transferring events over the distributed system.

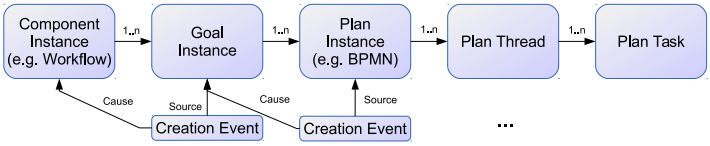


Fig. 3 Events can have both sources and causes based on the static component model and their runtime instances

Since the active component model uses a static component model, the component hierarchy can be used to denote both the source and the *cause* of an event (cf. Fig. 3). For example, a goal-oriented workflow component, when executed, consists of a component instance. If this component instance adopts a new goal instance, a creation event is issued. The source of this creation event is the newly-created goal instance, however, the cause of the event is the component instance which adopted the goal. Once the means-end reasoning decides on a plan to perform in order to reach the goal, a plan instance is created and another creation event is issued. In this case, the source is the plan instance while the cause of the event is the goal instance that triggered the means-end reasoning. This chain can be traced down the model hierarchy down to tasks within BPMN-based plans.

Since instances have unique identifiers, the events merely have to include the cause and source identifiers, minimizing the size of each event. Nevertheless, once the events have been gathered, the cause and effect chain can be recreated by cross-referencing them with other events. In addition, events that are delayed due to the distributed nature of the system can be identified and estimates can be given. For example, if the creation event of the plan instance is missing, the lower boundary for the creation time of the plan instance is the creation time of its cause, i.e. the goal instance. This can at the very least provide the user with an adequate approximation until the missing events arrive.

Each event also requires a timestamp, identifying when the event occurred, especially in relationship to other events. In distributed systems, simple timestamps often pose an issue since it is difficult to reliably synchronize clocks between nodes. Other approaches such as vector clocks increase both complexity and data volume. However, in the case of goal-oriented workflows as presented here, the problem is reduced since each individual workflow instance runs on a single node and its events are therefore internally consistent. Only if the cause chains crosses node barriers, for example due to service calls, caution has to be applied regarding synchronization. Nevertheless, small inconsistencies can be corrected to a certain degree using the same approach as mentioned above regarding missing events.

4 Monitoring Architecture and Implementation

The monitoring mechanism has been implemented using a monitoring service (IMonitoringService), which allows producers to publish events and consumers to subscribe for certain types of events. In Fig. 4 an overview of the architecture is depicted. It can be seen that in each platform a specific monitoring component realizes the monitoring service. Each component that creates (internally or intentionally) events, automatically publishes them to the corresponding service. For this purpose each component searches for an IMonitoringService when an event has to be published. In case a service is found the binding is fixed and the event is forwarded, if not the component stores the unavailability of the service and the event is dismissed. It will try to search for the service again when a new event occurs and after some specified time interval has elapsed. Event consumers can subscribe at the monitoring service using an optional event filter. This mechanism allows for reducing the network load as only events are transferred which pass the filter test.

The global monitoring infrastructure is set up as a peer to peer infrastructure formed by multiple monitoring components realizing an information exchange protocol. Knowing that events are reported locally from the event sources, each monitoring service searches for all other monitoring services and forwards local events to all remote services. In this way all monitoring services internally build up a globally consistent event state. For scalability reasons, the monitoring components only hold a certain amount of event in memory and dismiss older events. If longer lasting book keeping is necessary they can also be configured using a distributed database to store older events.

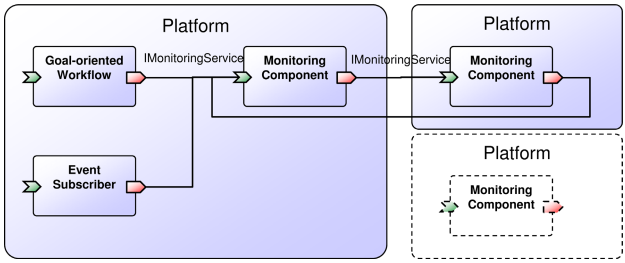


Fig. 4 Monitoring infrastructure

4.1 Workflow Environment

The monitoring service is further used to supplement a distributed workflow management system [11] implemented as active components.

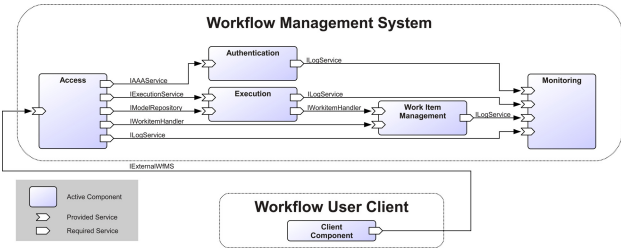


Fig. 5 Event types used over the event source life cycle

The system is largely derived from the Workflow Management Coalition Reference Model [7], where the monitoring service represents the monitoring subsystems of a traditional workflow management system. The workflow management system consists of multiple active components similar to the monitoring service which use service calls to exchange information.

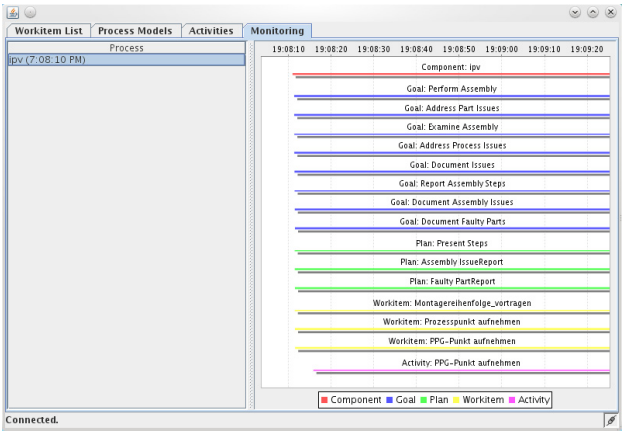


Fig. 6 Gantt chart monitoring of a goal-oriented workflow in the workflow client

Aside from the monitoring, other components are available to provide the rest of the workflow management functionality. The access component manages the access to workflow management functionality for workflow clients. The authentication component verifies the credentials of users and specifies their access rights. The execution component stores workflow models and launches workflow instances. Finally, the work item component holds work items representing human tasks for users to perform. Each component can be replicated to provide robustness and scalability.

The information gained by the monitoring component can be used to provide an overview of a goal-oriented workflow instance. As shown in Fig. 6, users can select a running or past workflow instance and is presented with a Gantt chart of event sources. The user can click on individual sources and is provided with a view which includes the clicked item and sources that were created as a result of the

item, providing a drilled-down view on parts of the process instance. If only partial information is available, an estimate can be shown to the user based on other event sources in the hierarchy. However, due to the chaining of references a subtree may be omitted if the cause-source hierarchy is interrupted. While this is remedied as soon as the missing information arrives, it is still a limitation of the system.

5 Evaluation and Outlook

In Section 1 we provided two areas where the system had to fulfill a set of requirements. The first area involved the cohesion between business goals and workflow actions. Here, the monitoring system is required to establish a relationship between the actions of the workflow and the business goals and allow the user to "drill down" from individual goals down to specific actions. The modeling of the goal-oriented workflows allow the events to establish a cause-source hierarchy to attribute actions to goals.

The first set of requirements was due to the required flexibility of a distributed workflow management system. The events may be generated by workflows anywhere within the distributed system, forwarded to interested nodes and the system should be robust and tolerate disappearing nodes. These three requirements were achieved by implementing the monitoring system as a event broker, forwarding all events it receives for publication to corresponding services. Unless a node is permanently disabled before it can transmit new events, all events will eventually be available to the workflow client. Low communication overhead was achieved by minimizing the amount of information stored in the events, including only references to sources and causes and letting the receiver reconstruct the chain. Furthermore, bulk events are available to bundle multiple events for transfer.

However, a remaining concern is the potential loss of all events of a specific event source. Due to the events containing only a minimum of information, this would interrupt a link between the main hierarchy tree and one of its subtrees. While there is a balance involved with regard to the event sizes, this challenge could be address by including a more information about the chain in each event by including not only the current causes but also a number of previous causes, allowing the workflow client to include the subtree and only omit the missing source.

References

1. Berry, M.J., Linoff, G.: Data Mining Techniques: For Marketing, Sales, and Customer Support. John Wiley & Sons, Inc., New York (1997)
2. Berson, A., Smith, S.J.: Data Warehousing, Data Mining, and Olap, 1st edn. McGraw-Hill, Inc., New York (1997)
3. Bratman, M.: Intention, Plans, and Practical Reason. Harvard University Press (1987)

4. Chaudhuri, S., Dayal, U.: An overview of data warehousing and olap technology. SIGMOD Rec. 26(1), 65–74 (1997)
5. Erol, K., Hendler, J., Nau, D.S.: Htn planning: Complexity and expressivity. In: AAAI, vol. 94, pp. 1123–1128 (1994)
6. Greiner, T., Düster, W., Pouatcha, F., von Ammon, R., Brandl, H.-M., Guschakowski, D.: Business activity monitoring of norisbank taking the example of the application easycredit and the future adoption of complex event processing (cep). In: Proceedings of the 4th International Symposium on Principles and Practice of Programming in Java, PPPJ 2006, pp. 237–242. ACM, New York (2006)
7. Hollingsworth, D.: Workflow Management System Reference Model. Workflow Management Coalition (1995)
8. Jander, K., Braubach, L., Pokahr, A., Lamersdorf, W.: Validation of Agile Workflows Using Simulation. In: Dastani, M., El Fallah Seghrouchni, A., Hübner, J., Leite, J. (eds.) LADS 2010. LNCS, vol. 6822, pp. 39–55. Springer, Heidelberg (2011)
9. Jander, K., Braubach, L., Pokahr, A., Lamersdorf, W., Wack, K.-J.: Goal-oriented processes with gpmn. International Journal on Artificial Intelligence Tools (IJAIT) 20(6), 1021–1041 (2011)
10. Jander, K., Lamersdorf, W.: Gpmn-edit: High-level and goal-oriented workflow modeling. In: WowKiVS 2011, vol. 37, pp. 146–157 (2011)
11. Jander, K., Lamersdorf, W.: Jadex WfMS: Distributed Workflow Management for Private Clouds. In: Conference on Networked Systems (NetSys), pp. 84–91. IEEE Xplore (2013)
12. Jennings, N., Norman, T., Faratin, P.: ADEPT: An agent-based approach to business process management. ACM SIGMOD Record 27(4), 32–39 (1998)
13. Object Management Group (OMG). Business Process Modeling Notation (BPMN) Specification, version 2.0 edition (January 2011)
14. Pietzuch, P.R., Bacon, J.: Hermes: A distributed event-based middleware architecture. In: Proceedings of the 22nd International Conference on Distributed Computing Systems, ICDCSW 2002, pp. 611–618. IEEE Computer Society, Washington, DC (2002)
15. Pokahr, A., Braubach, L.: The active components approach for distributed systems development. International Journal of Parallel, Emergent and Distributed Systems 28(4), 321–369 (2013)
16. 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)
17. Vassiliadis, P., Sellis, T.: A survey of logical models for olap databases. SIGMOD Rec. 28(4), 64–69 (1999)
18. Vassiliadis, P., Simitsis, A., Skiadopoulos, S.: Conceptual modeling for etl processes. In: Proceedings of the 5th ACM International Workshop on Data Warehousing and OLAP, DOLAP 2002, pp. 14–21. ACM, New York (2002)

Agent Based Negotiation of Decentralized Energy Production

Luca Tasquier, Marco Scialdone, Rocco Aversa, and Salvatore Venticinque

Abstract. The increasing demand for energy has stimulated the formulation of plans aiming at shifting to renewable energies, not only to save money, but also for the responsibility that the present population has towards future generations. Within this scenario, ICT solutions will help the citizens in saving energy: people of a community that have power's over-production can take benefits in terms of gain by selling the excess energy to neighbourhood instead of power suppliers due to the fees that they have to pay in the latter case. On the other hand, citizens that need additional energy can buy it from sellers in the neighbourhood at favorable prices. In this paper we present an agent-based framework that allows the collection of data about energy consumption in a neighbourhood and the negotiation of the whole produced/consumed energy among the involved parties in order to maximize the profits of the community.

Keywords: Multi-Agent Systems, Energy Market, Smart Grid.

1 Introduction

The increasing demand for energy has stimulated the formulation of plans aiming at shifting to renewable energies, not only to save money, but also for the responsibility that the present population has towards future generations. A combination of IT and telecommunication technologies is necessary to enable the saving of energy and resources: ICT based solutions will provide real time information on energy consumption in a home or a building, giving the possibility to citizens to take decisions in order to save energy, also by acquiring it from people of the neighbourhood that

Luca Tasquier · Marco Scialdone · Rocco Aversa · Salvatore Venticinque
Dipartimento di Ingegneria Industriale e dell'Informazione, via Roma 29, 81031 Aversa, Italy
e-mail: {luca.tasquier,marco.scialdone,rocco.aversa,
salvatore.venticinque}@unina2.it

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_8

have an energy's over-production and want to sell it; these people can take benefits in terms of gain by selling the energy to neighbourhood instead of power suppliers due to the fees that they have to pay in the latter case. In order to let the consumers to collaborate in using the local energy production and storage facilities in the neighbourhood, the integration with a variety of appliances in the buildings (e.g. solar panels, energy storage units, heating/cooling systems, etc.) is necessary. Therefore, it is needed the integration of information coming from many sources in order to implement negotiation among the producers and consumers, thus minimizing the exchange with power suppliers and maximizing the neighbourhood's gain.

In this paper we present an agent-based framework that allows the collection of data about energy consumption in a neighbourhood and the negotiation of the whole produced/consumed energy among the involved parties in order to maximize the profits of the community. The paper is organized as follows: Section 2 presents some related works; Section 3 describes the proposed architecture while in Section 4 we present the details of the framework's prototypal implementation; in Section 5 experimental results are given; finally conclusions are drawn in Section 6.

2 Related Work

Negotiation models for mapping consumption demand to produced energy have been widely investigated in literature in many fields; the complexity of an automated negotiation depends on several factors, such as the number of negotiated issues, dependencies among these issues, negotiation protocol, constraints, etc. Much effort has been spent on the investigation in the field of agents' technology [12]. In [11] authors consider how consumers might relate to future smart energy grids, and how exploiting software agents to help users in engaging with complex energy infrastructures. [10] proposes a Multi-Agent system architecture to simulate and analyze Competitive Electricity Markets combining bilateral trading with power exchange mechanisms. Several heterogeneous and autonomous intelligent agents representing the different independent entities in Electricity Markets are used: these agents have historical information about the market including past strategies of other players, and have strategic behaviours to face the market. [9] presents the architecture and negotiation strategy of an agent-based negotiation platform for power generating and power consuming companies in contract electricity market. An intelligent agent, by using fuzzy logic modification of Genetic Algorithm in order to accomplish strategy optimization, implements the negotiation process by selecting a strategy using learning algorithms. In [5] and [6] authors present an agent-based approach to manage negotiation among the different parties. The target is to use adaptive negotiation strategies in order to trade energy in a deregulated market; in order to optimize energy production and supply costs by means of negotiation and adaptation, strategies derived from game theory are used. In [15] another negotiation algorithm using game theory is proposed, where agents act on behalf of end users, thus implying the necessity of being aware of multiple aspects connected to the distribution of

electricity related to outside world variables like weather, stock market trends, location of the users etc. Authors also have previously experiences in building network of agents for negotiation and brokering of computational resources in Cloud markets [3, 7, 14].

The solution proposed in this paper focuses on coordinating local energy production and consumption of individual houses in a neighbourhood, thereby balancing the energy flow and reducing the fluctuations towards the central power grid: the agents make autonomous decisions on behalf of the users, trying to optimize his/her energy consumption; in this work we present, together with a negotiation strategy, a complete infrastructure that addresses the neighbourhood's information exchange, giving the possibility to use whatever technology in order to retrieve the data; furthermore the use of a distributed architecture makes the proposed framework scalable to larger neighbourhoods.

3 CoSSMic Architecture

CoSSMic (Collaborating Smart Solar-powered Micro-grids. FP7 - SMARTCITIES, 2013) is an ICT European project that aims at fostering a higher rate for self-consumption ($\succ 50\%$) of decentralized renewable energy production by innovative autonomic systems for the management and control of power micro-grids on users' behalf [2]. Micro-grids are spread in different neighborhoods. They embed renewable energy production, consumption, storage capacity and are combined with an intelligent ICT platform.

Each household of the CoSSMic community hosts a Home Area network (HAN) with a number of electrical devices. For each household a control agent (CA) manages the HAN in order to optimize self-consumption rates using renewable energy sources, and negotiates with other households to optimize the energy self-consumption within a neighborhood [1].

Energy usage is scheduled in accordance with policies defined by users, as well as other relevant information such as input from weather stations, weather forecasts, and habits and plans of inhabitants [2].

The proposed agent-based architecture is depicted in Figure 1.

The connection between devices and agents is implemented by a a distributed, or centralized, gateway *RESTFul Gateway* (RFG) that is in charge of notifying events using the agent common language (ACL) and message passing mechanisms. The gateway exposes a HTTP interface that wraps a REST protocol. The gateway receive incoming requests, verifies their correctness, supports the translation between the device protocols and agents' one, forwards the messages to the involved components within the platform. Two kinds of communication mechanisms are implemented by the gateway. *Asynchronous requests* are used to ask for some jobs to be scheduled. The gateway replies only with an acknowledge after that the asynchronous request has been received and its correctness has been verified. The response is provided by

the platform by a new connection, when the result is available. *Synchronous requests* are used to get any kind of information.

A specialized agent implements an *Event Bus* (EB) to support the internal delivery of incoming events by a a publish/subscribe paradigm. For each household of the neighborhood a *Control Agent* (CA) is in charge of controlling the building's devices scheduling their tasks according to house needing and to optimize energy consumption. CA can act both as a producer and as a consumer against the neighborhood, depending on whether the household needs to sell or to buy energy. Each time a new household join the neighborhood the Mediator Agent starts a new Control Agent.

The Control Agent registers itself to the Event Bus in order to receive messages related to the consumption or the production of energy within his building. After that its behavior is event-based. It start an defined handler for each event is notified by the event bus.

The *Protocol Handler* (PH) is the agent delegated to exchange energy by a defined negotiation protocol.

Different protocols can be loaded. It currently implements the FIPA Contract Net [13] and takes the role of *Initiator*. According to the Contract Net the Initiator is an agent that starts a negotiation by a *Call for Proposal* (CFP). If the initiator receives a proposal, it is evaluated and, according to the defined strategy, accepted or refused. If the initiator does not receive any proposals or none are accepted, it begins to behave as a*Responder*. It starts waiting for CFPs itself.

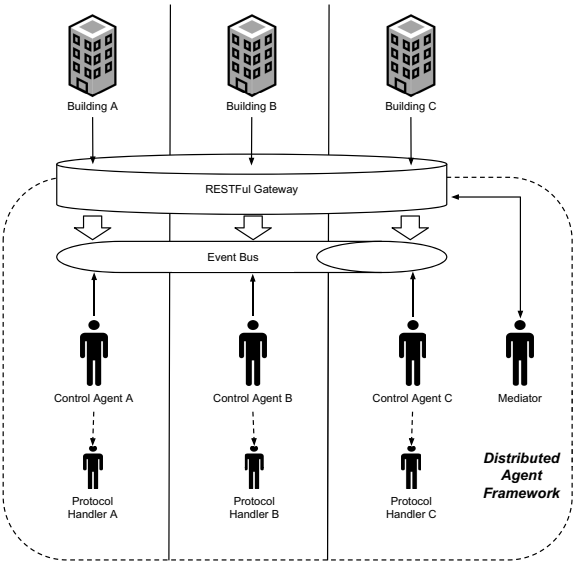


Fig. 1 Agent framework

4 Implementation

The proposed framework has been implemented by using JADE [4] as agent platform. Due to the adoption of this agent environment, the RESTFul Gateway has been conceived and implemented as a new *Message Transport Protocol* (MTP) for the agent platform. MTP is the mechanism that JADE offers to support inter-platform communication.

If a sensor wants to publish some information, it just sends a POST HTTP message with *http://$ip_address:$port/event/* as URL and a JSON string as message body. For each incoming HTTP-REST request, the gateway checks the correctness of the request url and verifies if all the mandatory headers are filled. After that it translates the request and forwards it to the related component of the CoSSMic architecture.

In order to configure all the RFG operations (request check, translation, forwarding) the Apache Derby database has been used [16]. In particular, three tables have been defined in order to support the RFG operations: the *URL Validation Table* is used to check if the request path is well-formed, the *URL Mapping Table* allows the verification of the request's correctness, the translation and the forwarding of the message, the *Header Validation Table* indicates the mandatory header fields for the URL Mapping Table requests. This approach made the extension of supported request easy due to the fact that in order to add a new supported message to the RFG it is only necessary to insert it into the framework's database, without managing the RFG source code.

The usage of a publish/subscribe middleware has been conceived to ensure scalability and high decoupling between the sensing part and the agent environment. Due to the relevant resource consumption and the not negligible installation effort of the existent publish/subscribe systems (such as queues, etc.), the Event Bus has been implemented as an agent within the framework. The EB Agent starts at the boot of the platform and provides mechanisms to publish events and forward them to the subscribers. The publishing occurs via the aforementioned REST request. The RFG forwards the message to the Event Bus that parses the event and forwards it to the subscribed agents. The event is represented by a JSON encoding [8]. It contains a number of information such as the house that generated the event, the event's type and its parameters. An example of JSON encoding is shown in Figure 2 (a). The Event Bus allows Control Agents to subscribe (and unsubscribe) themselves a multiple times and for the reception of specific types of events.

According to the event type the CA loads the correspondent handler. An event can inform about a the $power$ produced or consumed by a device. If the power value is positive, it means that energy is produced and then the CA will try to sell energy to the neighbourhood. Conversely, if the value is below 0, the energy was consumed and then the agent will look for producers to buy energy from the neighbourhood.

In our prototypal implementation, the negotiation strategy is very simple since the cost of energy is less than the one fixed by the power supplier (in particular we subtracted the cost of fees). It means that it is also greater than the cost paid by the

```
{
  "HouseID": "ID of house",
  "Date": "Date when event occurs",
  "Time": "Time when event occurs",
  "EventSource": "Event type",
  ... ,
  "Parameters": {
    "Power": "Power consumed/produced",
    ...
  }
}
```

(a) JSON Event

```
{
  "AgentId": "ID of agent",
  "Power": "Power to sell/buy",
  "Price": "Price at which to sell/buy",
  "Duration": "Contract duration",
  "Interruptible": "If the contract is
                    interruptible or
                    not",
  "Protocol": "Protocol to use",
  ...
}
```

(b) SLA JSON Message

Fig. 2 JSON Messages

provider to producers. By the way proposals from CoSSMic neighborhoud will be always preferred to the GenCO.

In this condition the only parameters that are evaluated during a negotiation are the amount of energy to buy/sell and the duration of the contract. Obviously, producers and consumers will purse complementary objectives:

- Consumers: try to get as much energy as it is required from the neighborhoud, thus achieving significant savings.
- Producers: try to sell to neighborhoud all the over production to CossMic consumer, to improve their income.

In our prototype the following cases have been handled:

1. if a consumer/producer cannot acquire/sell through a single negotiation all the energy, it can accept offers of the others until full satisfaction of their needs;
2. if a consumer cannot find enough energy within the neighbourhood, it will be forced to contact a GenCo;
3. if a producer fails to sell in a single transaction all the produced energy, it waits for new CFPs from other consumers;
4. if a consumer purchased a certain amount of energy but it does not use it all, it will try to resell the remaining energy (thus working as producer).

In order to establish a contract among the parties, a Service Level Agreement (SLA) has been used. These JSON messages are formatted according to the scheme shown in Figure 2 (b). The main elements are the power to sell/buy, the price at which to sell/buy, the duration of the contract and the protocol.

5 Energy Compensation and Expected Reward

In the experimental setup we deployed only one centralized gateway and all the agents are created on the same platform as it is drawn in Figure 1.

In order to evaluate feasibility and effectiveness of the proposed approach we used a synthetic workload built up by the reporting received from the energy provider of three houses and a swimming pool settled in the Province of Caserta.

The considered houses have not devices able to produce energy and thus they act as consumers. The swimming pool is equipped with a photo-voltaic system that allows the pool to produce more energy than is used, so serving as producer/consumer in our platform. In order to evaluate framework's performance, we also took into account the time-slots where the power supplier sells the energy at a higher cost (h-c) and a lower cost (l-c).

The workload is summarized in Table 1.

Table 1 Energy production and consumption summary

Building	**Production [kWh/year]**	**Consumption [kWh/year]**	**Consumed energy in h-c [%]**	**Consumed energy in l-c [%]**
House-01	0	2442	30.40	69.60
House-02	0	2681	34.50	65.50
House-03	0	2604	34	66
Swimming Pool	287100	248800	50	50

Unfortunately at the state of the art of the CoSSMic project monitoring infrastructures are not available at the trial sites yet. It means that we have not fine grained information neither about energy production and consumption during a day nor power requirements that can be assumed less than 3 kW. However, starting from the information in the report provided by the energy provider to their customers we were able to estimate, for each month, the average energy consumption and production in different time periods of the day.

The total costs and rewards are shown in Table 2 when there is not energy exchange between users. It has been computed using the tariffs of the provider that specify the cost per kWh for each time-slot in the different days of a week. The unit cost includes not only the energy cost but also some additional fees.

Table 2 Costs' summary without using our prototype

Building	**Total cost [EUR/year]**	**Reward [EUR/year]**
House-01	398.14	0
House-02	438.21	0
House-03	425.49	0
Swimming Pool	23826	17226

Let us observe what happens when the negotiation between producers and consumers is supported, and we set the price of the energy sold to the neighbors equal to the cost proposed by the GenCo minus the fees. In this case the three consumers always choose to buy energy from the swimming pool first. The swimming pool

gets a reward greater than the one received from the GenCo. The results we got are shown in Table 3.

Theoretically, if we suppose that the pools could store the energy and then sell it in the evening, the household savings is estimated at around 46%. Even if the pool produces only during the day and cannot store energy, the amount purchased at a lower price allows the buildings to save about 40%, and to use a relevant amount of green energy. Of course the number of buildings is too small and it does not allow to get a relevant benefit also for the swimming pool, which still sells most of the energy produced to GenCo. Moreover the data we have not allowed us to simulate the real scenario that can be affected by some fast dynamics not reconstructed by the poor input information.

Table 3 Costs' summary by using our prototype

Household	**From CoSSMic [kWh/year]**	**To CoSSMic [kWh/year]**	**From GenCo [kWh/year]**	**To GenCo [kWh/year]**	**Total cost [EUR/year]**
House-01	742	0	1700	0	248.66
House-02	925	0	1756	0	258.03
House-03	885	0	1719	0	252.32
Swimming Pool	0	2553	248800	284547	23774.95

6 Conclusion

In this paper we presented an agent-based framework that enables the management and negotiation of decentralized energy production in order to optimize energy usage and to save cost. We estimate the improvements in terms of greater utilization of green energy and cost savings by a simulated workload that has been computing using historical data belonging to three households and a swimming pool that hosts a big solar panel installations. Ongoing work is developing a monitoring infrastructure that will be able to collect information about energy consumption and production at different trial sites, to negotiate the energy exchange in a neighbourhood with a real workload that is characterized by a short sampling period and in real time.

Acknowledgements. This work has been supported by CoSSMic (Collaborating Smart Solar-powered Micro-grids - FP7-ICT-608806).

References

1. Amato, A., Martino, B.D., Scialdone, M., Venticinque, S.: An Agent-based Approach for Smart Energy Grids. In: Proceedings of the 6th International Conference on Agents and Artificial Intelligence, vol. 2, pp. 164–171 (2014)
2. Amato, A., Martino, B.D., Scialdone, M., Venticinque, S., Hallsteinsen, S., Horn, G.: Software Agents for Collaborating Smart Solar-powered Micro-grids. In: The Italian Association on Information System conference (itAIS 2013), vol. 2 (2014)
3. Aversa, R., Tasquier, L., Venticinque, S.: Cloud agency: A guide through the clouds. Mondo Digitale 13(49) (2014)
4. Bellifemine, F., Poggi, A., Rimassa, G.: JADE–A FIPA-compliant agent framework. Proceedings of PAAM 99, 33 (1999)
5. Capodieci, N., Alsina, E.F., Cabri, G.: A context-aware agent-based approach for deregulated energy market. In: 2012 IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), pp. 16–21. IEEE (2012)
6. Capodieci, N., Cabri, G., Pagani, G.A., Aiello, M.: Adaptive game-based agent negotiation in deregulated energy markets. In: 2012 International Conference on Collaboration Technologies and Systems (CTS), pp. 300–307. IEEE (2012)
7. Di Martino, B., Venticinque, S.: Distributed Agents Network for Ubiquitous Monitoring and Services Exploitation. In: 7th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC 2009), vol. 1, pp. 197–204. IEEE Computer Society, Washington, DC (2009), http://dx.medra.org/10.1109/CSE.2009.122, doi:10.1109/CSE.2009.122
8. Ihrig, C.J.: JavaScript Object Notation. In: Pro Node. js for Developers, pp. 263–270. Springer (2013)
9. Jia-hai, Y., Shun-kun, Y., Zhao-guang, H.: A multi-agent trading platform for electricity contract market. In: The 7th International Power Engineering Conference, IPEC 2005, pp. 1024–1029. IEEE (2005)
10. Praça, I., Ramos, C., Vale, Z., Cordeiro, M.: Intelligent agents for negotiation and game-based decision support in electricity markets. Engineering Intelligent Systems for Electrical Engineering and Communications 13(2), 147 (2005)
11. Rodden, T.A., Fischer, J.E., Pantidi, N., Bachour, K., Moran, S.: At home with agents: exploring attitudes towards future smart energy infrastructures. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1173–1182. ACM (2013)
12. Rogers, A., Ramchurn, S.D., Jennings, N.R.: Delivering the Smart Grid: Challenges for Autonomous Agents and Multi-Agent Systems Research. In: AAAI (2012)
13. Smith, R.G.: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. IEEE Transactions on Computers C-29(12), 1104–1113 (1980)
14. Venticinque, S., Tasquier, L., Di Martino, B.: Agents based cloud computing interface for resource provisioning and management. In: 2012 Sixth International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), pp. 249–256. IEEE (2012)
15. Whitehead, D.: The El Farol bar problem revisited: Reinforcement learning in a potential game. ESE Discussion Papers 186 (2008)
16. Zikopoulos, P.C., Baklarz, G., Scott, D.: Apache Derby–Off to the Races: Includes Details of IBM Cloudscape. IBM Press (2010)

Models of Autonomy and Coordination: Integrating Subjective and Objective Approaches in Agent Development Frameworks

Stefano Mariani, Andrea Omicini, and Luca Sangiorgi

Abstract. Objective and subjective approaches to coordination constitute two complementary approaches, which, being both essential in MAS engineering, require to be suitably integrated. In this paper, we *(i)* observe that a successful integration depends on the *models of autonomy* and *coordination* promoted by agent technologies, *(ii)* suggest that ignoring the two models may hinder agent autonomy, *(iii)* provide an example of "autonomy-preserving" integration by discussing TuCSoN4JADE.

1 Autonomy and Coordination: Issues

Autonomy is a core notion for agents as social entities in a multiagent system (MAS) [8]. *Interdependency* between agent activities is the foundation for the need for *coordination* [4], which becomes an essential facet of MAS design, as relevant as individual agent design [5]. Coordinating a society of agents towards the achievement of a social goal necessarily influences agent course of action, potentially hindering their autonomy—in particular when adopting the *objective* approach to the design of MAS coordination [13].

Objective coordination [13] refers to the coordination approaches – typical of the Software Engineering research field – where coordination-related concerns are extracted from agents to be embodied within dedicated abstractions – e.g., *coordination artefacts* [14] – offering *coordination as a service* to agents [23]. Coordination abstractions steer agent societies towards the achievement of social goals by managing dependencies between agent activities, even despite individual agent goals. On the other hand, *subjective coordination* [13] represents the dual approach, as it is typically adopted in the (Distributed) Artificial Intelligence field [10]. There, coordination issues are directly tackled by individual agents themselves, determining

Stefano Mariani · Andrea Omicini · Luca Sangiorgi
ALMA MATER STUDIORUM–Università di Bologna
via Sacchi 3, 47521 Cesena, FC, Italy
e-mail: {s.mariani,andrea.omicini}@unibo.it,
luca.sangiorgi6@studio.unibo.it

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_9

their best course of action in the attempt to achieve their own goals—which typically requires the intelligence to perform practical reasoning. As the result of agent own deliberation activity, subjective coordination basically expresses agent autonomy.

Thus, objective and subjective coordination clearly constitute two *complementary* approaches, both essential in MAS design and development [19], hence requiring to be suitably integrated—as also witnessed by many results already achieved. In [19], Activity Theory was proposed as the conceptual framework reconciling the objective and subjective approach, whereas in [16] TuCSoN coordination infrastructure [17] and JADE agent development framework [2] were integrated by offering TuCSoN tuple-based coordination as a JADE service. In [20], the CArtAgO framework [21] was integrated with three different agent platforms – Jason [3], 2APL [6] and simpA [22] – so as to enable and promote artefact-based interaction of heterogeneous agents.

In this paper, we first observe that the successful integration of objective and subjective coordination strongly depends on the technology level, that is, on the abstractions and mechanisms actually promoted by the agent frameworks. In particular, when building a MAS, integration depends on the *model of autonomy* promoted by the specific agent platform, and by its relationship with the *model of coordination* implemented by the specific (objective) coordination framework. Then, we show that any integration effort not taking into account such two aspects is likely to hinder agent autonomy by (unintentionally) creating *artificial dependencies* between the individual and the social stances on coordination. Finally, we provide an example of effective integration of objective and subjective coordination by discussing TuCSoN4JADE[1], where the model of autonomy promoted by the JADE platform is seamlessly integrated with the model of coordination provided by the TuCSoN middleware.

2 Autonomy and Coordination: Models and Technologies

Given the centrality of autonomy in the definition of agents, any agent development framework is required to provide architectural solutions to enable and support agent autonomy. Either explicitly or implicitly, such architectures assume what we call a *model of autonomy*, that is, a model defining *(i)* how agents behave as individual (autonomous) entities, *(ii)* how they relate to each other as social entities, as well as *(iii)* how the two things coexist. In Subsection 2.1 we analyse the model of autonomy promoted by two well-known agent development frameworks: JADE [2] and Jason [3].

In a similar way, the architectural components offering coordination services in agent infrastructures adhere to a *model of coordination*, which defines the semantics of the admissible interactions between agents in a MAS, in particular, w.r.t. their effects on the agent's control flow—hence, on agent autonomy. In Subsection 2.2 we

[1] Available at `http://bitbucket.org/smariani/tucson/downloads`

analyse the model of coordination provided by two well-known agent infrastructures: TuCSoN [17] and CArtAgO [21].

2.1 Autonomy and Coordination in Agent Development Frameworks

JADE

JADE (Java Agent DEvelopment Framework) [2] is a Java-based framework and infrastructure to develop open, distributed agent-based applications in compliance with FIPA standard specifications for interoperable, intelligent, multi-agent systems. In JADE, autonomy of agents is supported by the *behaviour* mechanism, whereas their mutual interaction depends on the *Agent Communication Channel* (ACC).

A behaviour can be logically interpreted as "an activity to perform with the goal of accomplishing a task". Thus, different "courses of actions" of a JADE agent are encapsulated into distinct behaviours the agent executes simultaneously. Technically, JADE behaviours are Java objects, which are executed *pseudo-concurrently* within a single Java thread by a *non-preemptive, round-robin scheduler*. During JADE agent initialisation, behaviours are added to the *ready queue*, ready to be scheduled. Then, method `action()` of the first behaviour – containing the agent's "course of action" – is executed. "Behaviours switch" occurs only when such method returns; hence, meanwhile *no other behaviour can start execution*—"non-preemptive" scheduler. Behaviour removal from the ready queue occurs only when the `done()` method returns `true`; otherwise, the behaviour is re-scheduled at the end of the queue—"round-robin" scheduler. Notice method `action()` is executed *from the beginning every time*: there is no way to "stop-then-resume" a behaviour.

The ACC is the run-time facility in charge of *asynchronous message passing* among agents: each agent has its own mailbox, and is notified upon reception of any message. JADE agents can communicate via several methods, among which:

`receive()` | to asynchronously retrieve the first message
`blockingReceive()` | to perform a *synchronous* receive

According to the JADE Programmers Guide [1], some care should be taken in using method `blockingReceive()`: in fact, it suspends *the agent*, not only the calling behaviour. This semantics impacts the aforementioned third dimension of the model of autonomy: "how the two things coexist". In fact, resorting to a synchronous communication mechanism hinders autonomy of the caller agent, since all its other behaviours – not just the caller one – are suspended by the communication semantics. In order to preserve agents autonomy, the JADE Programmers Guide suggests adoption of the following programming pattern: call `receive()` instead, then call method `block()` – of the `Behaviour` class – if no message is found, so as to let JADE suspend only the calling behaviour. The ubiquity of such pattern

in JADE code factually witnesses the relevance of the issue of understanding and suitably define the model of autonomy.

Summing up, JADE model of autonomy features *(i)* behaviours for individual tasks, *(ii)* asynchronous messages for subjective coordination, *(iii)* the "`block()`-then-resume" pattern to reconcile individual and social attitudes. Subsection 3.1 shows how any integration effort ignoring this semantics is bound to fail.

Jason

Jason [3] is both an agent language and an agent run-time system. As a language, it implements a dialect of AgentSpeak [18]; as a run-time system, it provides the infrastructure needed to execute a MAS. Although Jason is entirely programmed in Java, it features BDI agents, so a higher-level language (the Jason language) is used to program Jason agents using BDI abstractions. In Jason, autonomy of agents is supported by the Jason *plan/intention* execution machinery and the message passing facilities.

Like JADE behaviours, a Jason *plan* can be interpreted as a course of action to be performed to accomplish a task. Technically, a Jason plan differs considerably from JADE behaviours: *(i)* it is scheduled for execution as soon as a *triggering event* occurs, *(ii)* it is not directly executed "as is" (in general), but is instantiated as an *intention*, then executed, *(iii)* intentions are pseudo-concurrently executed *one action each*, according to a round-robin scheduler. Whereas in JADE the behaviour is the basic execution step, in Jason the same role is played by the single action, not by the plan/intention. Intentions may be *suspended* by the Jason reasoner, e.g. because the agent needs to wait for a message.

Jason agents can in fact exchange beliefs/plans/goals in the form of messages. Thus, subjective coordination is supported by these message passing facilities. In Jason, intentions are automatically suspended whenever they perform a "communication action" which cannot complete—to be resumed as soon as the action obtains its "completion feedback" (see [3], page 86). This preserves Jason agent autonomy similarly to behaviours in JADE: namely, by decoupling the control flow of a given "course of action" from the one of the agent undertaking them.

Summing up, Jason's model of autonomy features *(i)* plans/intentions for individual tasks, *(ii)* asynchronous message passing for subjective coordination, *(iii)* intention suspension mechanism to reconcile individual and social attitudes.

2.2 Autonomy and Coordination in Agent Infrastructures

TuCSoN

TuCSoN [17] is a Java-based, (logic) tuple-based coordination model and infrastructure for open, distributed MAS. It extends the LINDA model [7] by featuring

ReSpecT *tuple centres* [12] as its coordination artefacts [14], which are distributed over a network of TuCSoN nodes.

The TuCSoN architectural component that mostly explains its model of coordination is the *Agent Coordination Context* (ACC) [11]. ACCs are assigned to agents as they enter a TuCSoN-coordinated MAS to map coordination operations into events, *asynchronously* dispatching them to the coordination medium. Thus, ACCs are fundamental to guarantee and preserve agent autonomy: while the agent is free to choose and undertake its course of actions, its associated ACC takes care of communicating "coordination-related" events to TuCSoN—and of collecting results. In particular, ACCs enable separation of the *suspensive semantics* of a coordination operation from its *invocation semantics*. More precisely, the suspensive semantics implies that the operation itself is suspended if needed. Instead, the *synchronous* invocation semantics implies that *the agent, too* is suspended if the operation gets suspended.

To do so, every TuCSoN operation execution undergoes two steps:

invocation | the request to carry out a given coordination operation is sent to the TuCSoN tuple centre target of the operation

completion | the response to the coordination operation invoked is sent back to the requesting agent by the tuple centre

In other terms, any coordination operation in TuCSoN is *asynchronous by default*. Nevertheless, each of the TuCSoN coordination operations can be invoked either in a *synchronous* or in an *asynchronous* fashion—the agents choose.

Summing up, TuCSoN coordination paradigm preserves agent autonomy by decoupling the suspensive semantics of coordination operations from their invocation semantics, thanks to the ACC abstraction. In this way, synchronous calls are always consequences of the agent own deliberation process.

CArtAgO

CArtAgO [21] is a Java-based framework and infrastructure based on the A&A (agents & artefacts) meta-model [15]. A&A exploits *artefacts* as the tools that agents use to achieve their own goals—as humans do with their tools [9]. Artefacts can be used to uniformly represent any kind of environmental resource within a MAS—sensors, actuators, databases, etc.

Even though CArtAgO does not focus on coordination, its general-purpose artefacts programming model allows coordination artefacts to be designed. Thus, a model of coordination can be devised, in particular, based on the *agent body* abstraction. By exposing an *effectors* API and a *perception* API, CArtAgO agent bodies are the architectural components enabling (and decoupling) agent interactions with artefacts. By exploiting the effectors API, current agent activity is *suspended* until an event reporting the action *completion* is received: then, the corresponding activity resumed. Even if one activity is suspended, the agent *is not*: its working

cycle can continue processing percepts and executing other actions related to other activities.

Mediation by agent bodies is the mechanism preserving agent autonomy in CArtAgO by uncoupling action suspension from caller agent suspension.

3 Autonomy-Preserving Integration Approaches

This section tackles the issue of preserving agent autonomy when integrating objective and subjective coordination at both the conceptual level – according to the models of autonomy and coordination – and the technological level [16, 20].

In [20], CArtAgO is integrated with three different agent development frameworks: Jason [3], 2APL [6] and simpA [22]. There, CArtAgO is proposed as a framework to enable and promote artefact-based interaction of heterogeneous agents. Nevertheless, authors *de facto* integrate subjective and objective coordination: in fact, by allowing Jason, 2APL, and simpA agents to exploit CArtAgO artefacts, they make it possible to build & use coordination artefacts, effectively integrating the message-based (subjective) coordination capabilities of agents with the artefact-based (objective) ones. The approach taken in [20] is an example of *autonomy-preserving integration*: e.g., in the case of Jason-CArtAgO, Jason intentions suspension mechanism is successfully integrated with CArtAgO artefacts by exploiting CArtAgO agent body abstraction. In particular, whenever a Jason agent requests execution of an operation on a CArtAgO artefact, the caller intention is automatically suspended until the "effector feedback" is received. Thus, nothing can hinder Jason agent autonomy if they *simultaneously* operate on artefacts while exchanging messages with other agents.

In [16], integration between JADE and TuCSoN technologies is successfully achieved, allowing JADE agents to exploit TuCSoN coordination services as part of the JADE platform—however, without preserving autonomy. JADE model of autonomy and TuCSoN model of coordination were not considered: in fact, if a coordination operation gets suspended, the caller behaviour is unavoidably suspended, too, because of its single thread of control being stuck waiting for operation completion. This inevitably leads to the suspension of all other behaviours the agent is (possibly) concurrently executing. Roughly speaking, the agent choice to rely on objective coordination may affect its ongoing subjective coordination activities. This is a clear example of an artificial dependency (unintentionally) created by a "non autonomy-preserving" approach—as [16] is.

In the remainder of this section, an autonomy-preserving integration called TuCSoN4JADE is presented, which successfully solves such an issue.

3.1 Preserving Autonomy in TuCSoN4JADE

The first step to integrate TuCSoN and JADE is to implement TuCSoN as a JADE *service*, actually following the work in [16]. The main novelty here concerns the `BridgeToTucson` class, as the component mediating all the interactions between JADE and TuCSoN. In particular, it offers two methods for invoking coordination operations, one for each *invocation semantics* JADE agents may choose:

`synchronousInvocation()` | lets agents invoke TuCSoN coordination operations *synchronously w.r.t. the caller behaviour*. This means the caller behaviour *only* is (possibly) suspended – and automatically resumed – as soon as the requested operation completes, not the agent as a whole—as in [16].

`asynchronousInvocation()` | lets clients *asynchronously* invoke TuCSoN coordination operations. Regardless of whether the coordination operation suspends, the agent does not, thus the caller behaviour continues.

Fig. 1 shows what happens when a synchronous operation is invoked — asynchronous invocation is not so interesting for the purpose of the paper. The "alt"-labelled frame enclosing "JADE Behaviour" entity represents the equivalent of JADE "`block()`-then-resume" programming pattern in TuCSoN4JADE. In particular, once the synchronous invocation is requested (message 2), two scenarios may occur:

completion ready | the TuCSoN operation completion event has already been generated by the TuCSoN middleware, and is already available for inspection within TuCSoN4JADE bridge (messages 3.a-4.a)

operation pending | the completion event has not reached the `BridgeToTucson` object yet—thus, from TuCSoN4JADE standpoint, the invoked operation is still pending (messages 3.b-9)

In the second case, the behaviour blocks (step 3b), waiting to be automatically resumed by TuCSoN4JADE as soon as the operation completion becomes available. Meanwhile, `BridgeToTucson` delegates execution of the operation to its associated ACC. Once such operation is finally completed, steps 7-9 cause the caller behaviour to resume. Back to the first case, we understand how TuCSoN4JADE autonomy-preserving integration technically works. We know when JADE behaviour is re-scheduled, its `action()` method re-starts *from the beginning*, thus, method `synchronousInvocation()` is re-invoked. The whole TuCSoN4JADE machinery works because such method internally (thus transparently) checks if the completion of the operation just invoked is already available: only if it is not, the whole path 3.b-9 is executed. In case it is available, `BridgeToTucson` immediately sends completion event back to the caller behaviour (step 3a).

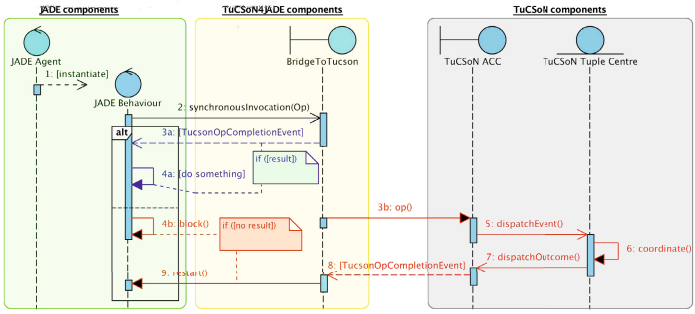


Fig. 1 TuCSoN4JADE *autonomy-preserving integration*, allowing JADE model of autonomy and TuCSoN model of coordination to integrate

3.2 Showcasing TuCSoN4JADE*: The "Book Trading" Example*

The "book trading" example is included in JADE distribution to showcase support to FIPA interaction protocols—thus, subjective coordination. In short, n seller agents advertise their catalogue of books, whereas m buyer agents browse such catalogues looking for books. The whole interaction takes the form of the well-known ContractNet protocol: buyers start a call-for-proposals, sellers reply with actual proposals, buyers choose which one to accept, the purchase is carried out. A fundamental requirement is that sellers should stay reactive to call-for-proposals even in the middle of a purchase transaction—otherwise they could lose potential revenues. We call *concurrency property* such a requirement. In the following, we take the book trading example as a paradigmatic example showcasing the practical relevance of autonomy-preserving integration approaches.

In particular, we re-think the ContractNet protocol by integrating objective and subjective coordination approaches: tuple-based call-for-proposals with message-based purchase. In fact, since the call-for-proposals should reach all the sellers, it is more efficient to put a single "call-for-proposals tuple" in a shared "contract-net space", rather than messaging each seller individually. On the contrary, since the purchase is typically a 1-to-1 interaction, messaging can efficiently do the job. This is not only conceptually correct, but also is more efficient – less messages, less network operations, etc. – in integrating an objective approach to coordination with a subjective one. We do so first exploiting the integration of TuCSoN and JADE proposed in [16] (Fig. 2), then using TuCSoN4JADE[2] (Fig. 3): in the former case, the concurrency property – thus, agent autonomy – is lost, whereas in the latter it is preserved as expected.

Fig. 2 depicts one possible instance of the run-time interactions between a given seller and a given buyer. In particular, the seller is replying to a previous call-for-

[2] The code is available as part of the TuCSoN4JADE distribution, downloadable from `http://bitbucket.org/smariani/tucson/downloads`

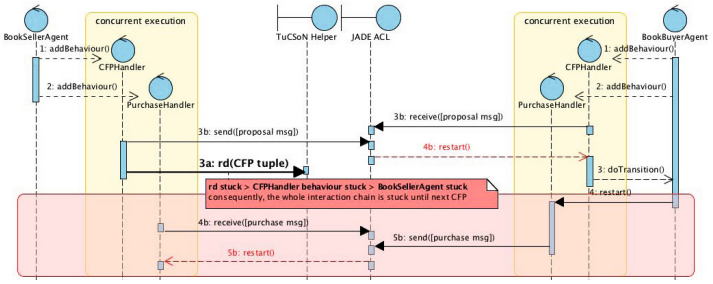


Fig. 2 Non autonomy-preserving approach taken in [16]: `rd` suspensive semantics extends to the caller behaviour, then to the caller agent, blocking all its activities

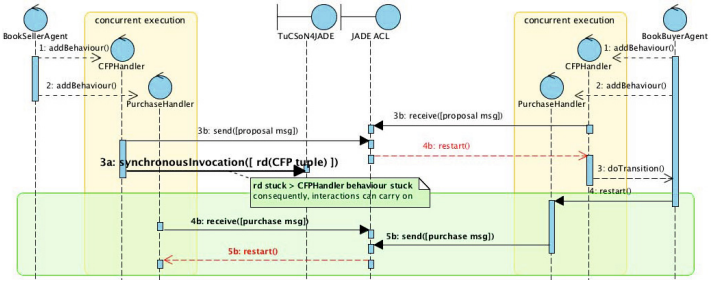


Fig. 3 TuCSoN4JADE autonomy-preserving approach: `rd` suspensive semantics is confined to the caller behaviour only, then the caller agent can carry on its other activities

proposals (message 3b). Meanwhile, it is also ready to serve new incoming call-for-proposals (3a). Here is the problem: the suspensive coordination operation `rd` gets stuck until a call-for-proposals is issued by a buyer. This is fine: it is exactly for this suspensive semantics that the LINDA model works. What is not so fine is the non autonomy-preserving approach taken by the `TucsonHelper` class in [16]: the `rd` is stuck on a network-level call and no "defensive" programming mechanism has been implemented to shield the caller behaviour. Thus it is stuck too, hindering the caller agent from scheduling other behaviours in the meanwhile—in particular, the "purchase" interaction chain (4b-5b) cannot carry on until a new call-for-proposals is issued.

Fig. 3 depicts the same scenario programmed upon the TuCSoN4JADE bridge, preserving autonomy. Since the `rd` call is shielded by a proper mechanism within the bridge, the suspensive semantics is confined to the caller behaviour. This means that only the caller behaviour is suspended – using the proper mechanisms provided by JADE, e.g. method `block()` – whereas other activities can carry on concurrently—e.g., the purchase transaction already in place (4b-5b).

The general applicability of the ContractNet protocol and its suitability for implementation as an "hybrid" protocol, drawing from both objective and subjective approaches, makes a correct integration of the two even more relevant in the context of agent development frameworks and coordination technologies.

4 Conclusion

Our goal is not just to show how JADE and TuCSoN were better integrated w.r.t. [16]—technically, it may even be seen as just a smarter implementation of the well-known OO "bridge pattern". Instead, we aim at stressing how technology-level details may have deep consequences on the higher levels of abstraction, whenever the models (possibly implicitly) brought about by technologies are not properly accounted for and understood. In particular, we demonstrate how the models of autonomy and coordination promoted by agent development frameworks may hamper an essential feature of agents: autonomy. Even though we discussed just a few agent-oriented frameworks, the issue of autonomy-preserving approaches in integrating subjective and objective coordination is quite a general one—thus, further work will be devoted to analyse other frameworks.

References

1. Bellifemine, F., Caire, G., Trucco, T., Rimassa, G.: Jade programmer's guide. Jade version 3 (2002)
2. Bellifemine, F.L., Poggi, A., Rimassa, G.: JADE–a FIPA-compliant agent framework. In: 4th International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM 1999), April 19-21, pp. 97–108. The Practical Application Company Ltd., London (1999)
3. Bordini, R.H., Hübner, J.F., Wooldridge, M.J.: Programming Multi-Agent Systems in AgentSpeak using Jason. John Wiley & Sons (October 2007)
4. Castelfranchi, C., Cesta, A., Conte, R., Miceli, M.: Foundations for interaction: The dependence theory. In: Torasso, P. (ed.) AI*IA 1993. LNCS, vol. 728, pp. 59–64. Springer, Heidelberg (1993)
5. Ciancarini, P., Omicini, A., Zambonelli, F.: Multiagent system engineering: The coordination viewpoint. In: Jennings, N.R. (ed.) ATAL 1999. LNCS (LNAI), vol. 1757, pp. 250–259. Springer, Heidelberg (2000)
6. Dastani, M., Meyer, J.-J.C.: A practical agent programming language. In: Dastani, M., El Fallah Seghrouchni, A., Ricci, A., Winikoff, M. (eds.) ProMAS 2007. LNCS (LNAI), vol. 4908, pp. 107–123. Springer, Heidelberg (2008)
7. Gelernter, D.: Generative communication in Linda. ACM Transactions on Programming Languages and Systems 7(1), 80–112 (1985)
8. Hexmoor, H., Castelfranchi, C., Falcone, R. (eds.): Agent Autonomy, Multiagent Systems, Artificial Societies, and Simulated Organizations, vol. 7. Springer (2003)
9. Nardi, B.: Context and Consciousness: Activity Theory and Human-computer Interaction. MIT Press (1996)
10. O'Hare, G.M., Jennings, N.R. (eds.): Foundations of Distributed Artificial Intelligence. Sixth-Generation Computer Technology. John Wiley & Sons (April 1996)

11. Omicini, A.: Towards a notion of agent coordination context. In: Marinescu, D.C., Lee, C. (eds.) Process Coordination and Ubiquitous Computing, ch. 12, pp. 187–200. CRC Press, Boca Raton (2002)
12. Omicini, A., Denti, E.: From tuple spaces to tuple centres. Science of Computer Programming 41(3), 277–294 (2001)
13. Omicini, A., Ossowski, S.: Objective versus subjective coordination in the engineering of agent systems. In: Klusch, M., Bergamaschi, S., Edwards, P., Petta, P. (eds.) Intelligent Information Agents. LNCS (LNAI), vol. 2586, pp. 179–202. Springer, Heidelberg (2003)
14. Omicini, A., Ricci, A., Viroli, M.: Coordination artifacts as first-class abstractions for MAS engineering: State of the research. In: Garcia, A., Choren, R., Lucena, C., Giorgini, P., Holvoet, T., Romanovsky, A. (eds.) SELMAS 2005. LNCS, vol. 3914, pp. 71–90. Springer, Heidelberg (2006)
15. Omicini, A., Ricci, A., Viroli, M.: Artifacts in the A&A meta-model for multi-agent systems. Autonomous Agents and Multi-Agent Systems 17(3), 432–456 (2008)
16. Omicini, A., Ricci, A., Viroli, M., Cioffi, M., Rimassa, G.: Multi-agent infrastructures for objective and subjective coordination. Applied Artificial Intelligence 18(9-10), 815–831 (2004)
17. Omicini, A., Zambonelli, F.: Coordination for Internet application development. Autonomous Agents and Multi-Agent Systems 2(3), 251–269 (1999)
18. Rao, A.S.: AgentSpeak(L): BDI agents speak out in a logical computable language. In: Perram, J., Van de Velde, W. (eds.) MAAMAW 1996. LNCS, vol. 1038, pp. 42–55. Springer, Heidelberg (1996)
19. Ricci, A., Omicini, A., Denti, E.: Activity Theory as a framework for MAS coordination. In: Petta, P., Tolksdorf, R., Zambonelli, F. (eds.) ESAW 2002. LNCS (LNAI), vol. 2577, pp. 96–110. Springer, Heidelberg (2003)
20. Ricci, A., Piunti, M., Acay, L.D., Bordini, R.H., Hübner, J., Dastani, M.: Integrating artifact-based environments with heterogeneous agent-programming platforms. In: 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), May 12-16, pp. 225–232. IFAAMAS, Estoril (2008)
21. Ricci, A., Viroli, M., Omicini, A.: CArtAgO: A framework for prototyping artifact-based environments in MAS. In: Weyns, D., Van Dyke Parunak, H., Michel, F. (eds.) E4MAS 2006. LNCS (LNAI), vol. 4389, pp. 67–86. Springer, Heidelberg (2007)
22. Ricci, A., Viroli, M., Piancastelli, G.: simpA: A simple agent-oriented Java extension for developing concurrent applications. In: Dastani, M., El Fallah Seghrouchni, A., Leite, J., Torroni, P. (eds.) LADS 2007. LNCS (LNAI), vol. 5118, pp. 261–278. Springer, Heidelberg (2008)
23. Viroli, M., Omicini, A.: Coordination as a service. Fundamenta Informaticae 73(4), 507–534 (2006)

Distributed Runtime Verification of JADE Multiagent Systems

Daniela Briola, Viviana Mascardi, and Davide Ancona

Abstract. Verifying that agent interactions in a multiagent system (MAS) are compliant to a given global protocol is of paramount importance for most systems, and is mandatory for safety-critical applications. Runtime verification requires a proper formalism to express such a protocol, a possibly non intrusive mechanism for capturing agent interactions, and a method for verifying that captured interactions are compliant to the global protocol. Projecting the global protocol onto agents' subsets can improve efficiency and fault tolerance by allowing the distribution of the verification mechanism. Since many real MASs are based on JADE, a well known open source platform for MAS development, we implemented a monitor agent that achieves all the goals above using the "Attribute Global Types" (AGT) formalism for representing protocols. Using our JADE monitor we were able to verify FYPA, an extremely complex industrial MAS currently used by Ansaldo STS for allocating platforms and tracks to trains inside Italian stations, besides the Alternating Bit and the Iterated Contract Net protocols which are well known in the distributed systems and MAS communities. Depending on the monitored MAS, the performances of our monitor are either comparable or slightly worse than those of the JADE Sniffer because of the logging of the verification activities. Reducing the log files dimension, re-implementing the monitor in a way independent from the JADE Sniffer, and heavily exploiting projections are the three directions we are pursuing for improving the monitor's performances, still keeping all its features.

1 Introduction

Verification of the compliance of interaction protocols in distributed and dynamic systems is of paramount importance for most applications. This can take place at design-time (offline or static verification) or at runtime (online or dynamic). In the latter case, a layer between the monitor executing the verification and the system

Daniela Briola · Viviana Mascardi · Davide Ancona
DIBRIS, Genoa University, Italy
e-mail: {daniela.briola,viviana.mascardi,davide.ancona}@unige.it

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_10

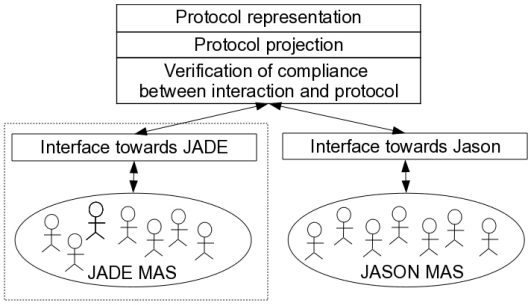


Fig. 1 Our modular framework for distributed runtime verification of MASs

under monitoring must exist, so that interactions can be captured and verified against the protocol.

If the system has been engineered as a multiagent system (MAS), then the choice of JADE[1] as the platform for implementing it may be a very natural one. JADE, implemented in Java, is the state-of-the-art tool for MAS development and has been used for many real industrial applications, as described in the JADE Homepage. FYPA (Find Your Path, Agent! [6–8]) is another industrial MAS developed in JADE and currently being used by Ansaldo STS, the Italian leader in railways signaling and automation, for allocating platforms and tracks to trains inside Italian stations in quasi-real time. Many academic applications spanning different domains are also described in the literature ([4, 13], just to cite a few ones). Due to the wide range of possible application fields and to the large amount of real use cases of JADE, supporting runtime verification of interaction protocols in JADE MASs would be a concrete step towards the reliability reinforcement and the industrial exploitation of MASs: in this paper we describe our contribution for the achievement of this goal.

We have designed and implemented a framework for distributed runtime verification of MASs and a dedicated layer for monitoring JADE interactions. The framework consists of **(1)** a formalism for describing "agent interaction protocols" (AIPs) based on Attributes Global Types (AGT) [1, 10]; **(2)** an algorithm to project AIPs onto subsets of agents, to obtain simpler protocols expressed in the same AGT formalism [2]; **(3)** a mechanism for capturing messages between the JADE agents under monitoring, in a transparent way; and **(4)** a method for verifying that interactions are compliant with the AIP [3].

The strength of our framework, represented in Figure 1, is its high modularity and potential for code reuse, because the first three components are independent from the actual MAS under monitoring. The fourth one (in a dashed box in the figure) is the subject of this paper, and has been expressly developed for JADE. A layer has been developed for Jason[2] too [3].

The paper is organized as follows: Section 2 describes the design and implementation of the JADE monitor; Section 3 describes the three MASs we have monitored

[1] `http://jade.tilab.com`

[2] `http://jason.sourceforge.net`

in order to assess the feasibility of our proposal, Section 4 describes our experiments and presents a performance analysis, and Section 5 discusses related approaches and concludes.

2 Runtime Verification of JADE MASs

In order to verify at runtime the interactions taking place in a JADE MAS, we have designed a monitor meeting the following requirements for non intrusiveness and code reuse:

1. the monitor must be able to supervise the execution of the MAS *without interfering with it*,

2. the monitor activity must require *no changes to the agents' code*,

3. the formalism for representing the AIP must be *AGT*,

4. the Prolog code already developed for implementing verification of interactions w.r.t. AGT and for protocol projection *must be re-used as it is*.

To meet requirements 1 and 2 we extended the JADE Sniffer agent, which is able to capture all the messages exchanged during the execution of the MAS in a non intrusive way: JADE is developed under the LGPL (Lesser General Public License) and the Java source code is available to the research community, so we were able to modify it to achieve our goals.

To meet requirements 3 and 4 we exploited the JPL library[3], providing a bidirectional interface between Java and SWI Prolog. As all our previous works on AGT were implemented in Prolog, allowing our JADE Monitor to use Prolog programs and predicates was the best way to ensure reusability.

The monitor is sketched in Figure 2 and is highly modular: we modified the code of the JADE Sniffer's class just as little as possible and we defined the method which converts a JADE message into a Prolog representation amenable for runtime verification in a separate class, to allow developers to modify that class only if a parsing different from the one we provided is required.

The monitor reads a file containing the Prolog code implementing verification and projection, and a configuration file listing the agents to be monitored, and onto which the protocol projection will be performed. A log file is written as the monitoring goes on.

The Prolog file contains definitions for three predicates:

– `initialize(LogFile, SniffedAgents)`, which sets `LogFile` as the file where writing the outcome of the verification, and projects the global protocol defined by the `global_type/1` predicate onto `SniffedAgents`.

– `remember(ParsedMsg)`, which inserts the Prolog representation of the JADE captured message into a message list, where messages are ordered by time stamp (if they have a time stamp, which is not mandatory) or in order of arrival.

– `verify(CurrentTime)`, which verifies the compliance to the global protocol of each message remembered in the message list and whose time stamp is lower than `CurrentTime`.

[3] `http://www.swi-prolog.org/packages/jpl/java_api/`

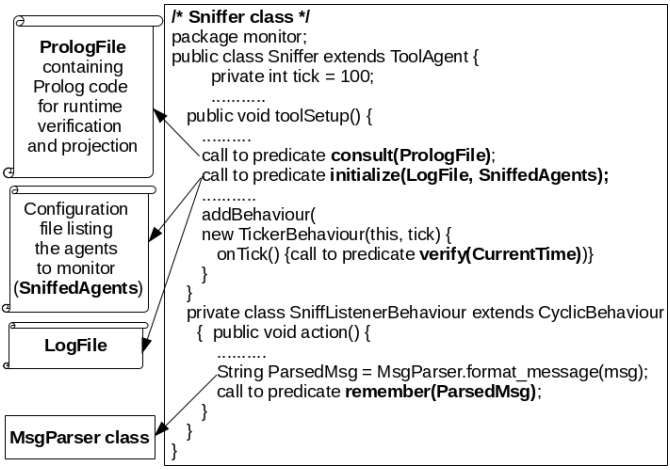


Fig. 2 The JADE monitor

These predicates are called in different methods of the monitor code:

– in the `toolSetup()` method, which initializes the agent, the Prolog file is consulted to make the predicates defined there available, and the `initialize(LogFile, SniffedAgents)` predicate is called;

– in the `action()` method of the `SniffListenerBehaviour` class, the JADE message `msg` is translated into a Prolog term by calling `ParsedMsg = MsgParser.format_message(msg)` and the obtained term is saved into the Prolog message list by calling `remember(ParsedMsg)`;

– a new `Ticker` behavior, re-executed every `tick` milliseconds (`tick` is set to 100 in our setting) is added to the monitor in the setup. This behavior calls the predicate `verify(CurrentTime)`, so that every 100 milliseconds all the messages exchanged in the last 100 milliseconds are verified.

The choice of first remembering the captured messages, and then verifying them, is due to problems with the order in which messages are forwarded to the JADE Sniffer agent, that sometimes do not respect their actual order: if this happens, the monitor could identify a violation of the protocol due to the wrong order of messages when, actually, the violation does not exist. To avoid this risk, we decided to split the interaction verification into two phases. In this way no problems due to the captured messages order arise, provided that the capturing delay is lower than the `tick` value. On the other hand, a violation of the protocol could be identified some milliseconds later, because messages are not checked as soon as they arrive. Our choice of delaying the violation identification rather then raising false violations can be easily changed calling the `verify` predicate as soon as a message is received, after the call to the `remember` predicate. The log file (the excerpt below refers to the Alternating Bit Protocol mentioned in Section 3) stores the result of parsing and verification in the form:

```
Conversion from Jade message (INFORM
  :sender(agent-identifier:name bob@...  :addresses(sequence ...))
  :receiver(set(agent-identifier:name carol@...  :addresses (...)))
  :content  "m2")
```

```
to Prolog message
  msg(performative(inform),sender(bob),receiver(carol),content(m2))
which leads from protocol state
(m2:m3:m1:**|(a1, 0):(m1, 1):**)|(m2, 1):(a2, 0):**|(m3, 1):(a3, 0):**
to protocol state
(m3:m1:m2:**|(a1, 0):(m1, 1):**)|(a2, 0):(m2, 1):**|(m3, 1):(a3, 0):**
```

Messages are also printed on the shell, for getting an immediate feedback on the MAS execution.

3 Test Cases

By means of AGT we were able to concisely represent protocols which are well known in the concurrent systems and MAS communities, like the Alternating Bit Protocol (ABP[4]) and the FIPA Iterated Contract Net Protocol (ICNP[5]). We developed two MASs that are expected to adhere to these protocols, in order to verify the ability of our monitor to detect deviations from the expected behavior and to assess its performances.

Our instance of the ABP MAS involves one agent `bob` that sends `m1` to `alice`, `m2` to `carol`, `m3` to `dave`, and waits for their respective acknowledges `a1`, `a2`, `a3` before resending `m1`, `m2`, `m3`, with the constraint that for each iteration i, $\texttt{m1}_i$ must precede $\texttt{m2}_i$, which must precede $\texttt{m3}_i$, and each acknowledge $\texttt{a}k_i$ must follow $\texttt{m}k_i$ and precede $\texttt{m}k_{i+1}$, with k ranging from `1` to `3`.

The ICNP MAS exploits the JADE implementation of the ICNP FIPA protocol offered by the `jade.proto` package[6] and one implementation of the ICNP MAS provided by JADE's developers[7]: in our instance, one `sender` agent playing the role of Initiatior interacts with three `receivers` playing the role of Responder, numbered from `1` to `3`.

The representation of the ABP and ICNP protocols using our AGT formalism is described in [10], where the advantages in terms of readability and conciseness with respect to other existing proposals are widely discussed. Due to space constraints, the reader is invited to refer to [10] for more details.

The FYPA (Find Your Path, Agent!) MAS was developed in JADE starting from 2009. It automatically allocates trains moving into a railway station to tracks, in order to allow them to enter the station and reach their destination (either the station's exit or a node where they will stop) considering real time information on the traffic inside the station and on availability of tracks. The station can be modeled as a direct non planar graph, where nodes are special railway tracks where trains can stop, and arcs are railway tracks connecting two nodes. The FYPA Reservation protocol described in AUML in Figure 3 involves agents representing trains and nodes. Each train knows the paths $\{P_1 = N_s...N_e; ...; P_k = N_s...N_e\}$ it could follow to go from the node where it is (N_s, for *start*), to the node where it needs to stop (N_e, for *end*).

[4] `en.wikipedia.org/wiki/Al-ter-na-ting_bit_protocol`

[5] `fipa.org/specs/fipa00030`

[6] `http://jade.tilab.com/doc/api/jade/proto/package-summary.html`

[7] `http://jade.tilab.com/doc/examples/protocols.html`

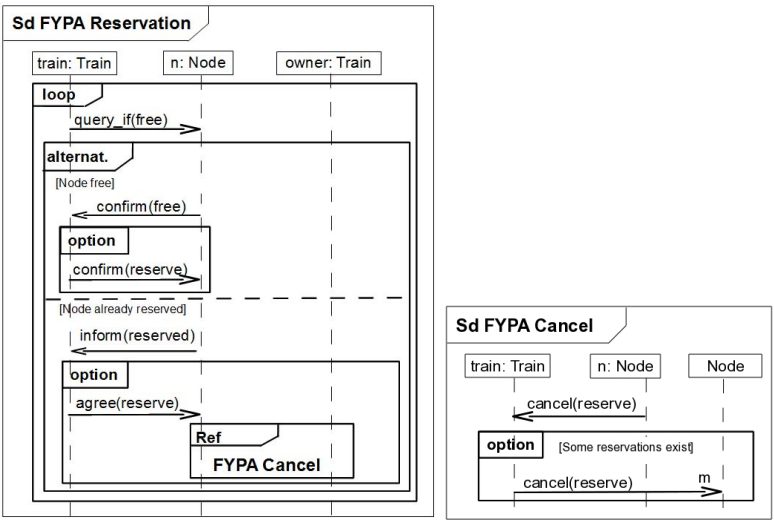


Fig. 3 FYPA Reservation protocol

Such paths are computed by a legacy Ansaldo application which is wrapped by an agent named *PathAgent*, not modeled here. Each train also knows which path it is currently trying to reserve, how many nodes answered to its requests and in which way, and how much delay it can accept: to reserve a path, the train must obtain a reservation for each node in it. To reserve a node, a train T_1 asks if it is free, waits for the answer from the node (free or already reserved by another train in an overlapping time slot) and then reserves the resource, which might also mean stealing it to the train T_2 that reserved it before (this usually takes place if the priority of T_1 is higher than that of T_2). In this case, the node will inform train T_2 by following the Cancel protocol, and T_1 will try to reserve the same path in different time slots. Each node knows the arcs that it manages (those that enter in it). It also knows which trains optioned or reserved the node, in which time slots, from which node they are expected to arrive, and which arc they can traverse.

In [11] we presented the formalization of the FYPA protocol using AGTs. That formalization, with minor modifications, has been used for verifying the MAS actual executions as discussed in Section 4.

In the instance of FYPA we tested, train `treno_1` tries to reserve the path `nodo_1, nodo_3, nodo_4, nodo_6` under the following conditions:
– FYPA1: all the nodes in the path are free, as there were no previous reservations: the reservation is completed without any problem;
– FYPA2: there was a previous reservation for one node (`nodo_3`), by a train with priority higher than `treno_1`'s priority: `treno_1` must change the reservation slots for its path;

– FYPA3: there was a previous reservation for one node (`nodo_3`), by a train with priority lower than `treno_1`'s priority: `treno_1` steals the reservation and successfully reserves the full path.

4 Experiments

We tested our JADE monitor with the ABP, the ICNP, and FYPA. With the ABP, which does not use attributes, we were also able to successfully check that projection works as expected. The results were the expected ones in case of both absence and presence of protocol violations.

Because of space constraints, we cannot provide details on all the three MASs. In this section we give the flavor of which kind of properties we were able to test with the FYPA MAS.

The test station consists of six nodes and train `treno_1`, with priority 2, enters from `nodo_1` and then moves to `nodo_3, nodo_4, nodo_6`.

The AGT modeling the FYPA protocol has been described in [11]. As discussed below, we were able to test "local", "horizontal" and "vertical" properties of messages. All our tests gave the expected result, namely a violation was correctly detected when we manually inserted some error in the message content or order, and the protocol verification correctly terminated when we did not insert any error.

"Local" properties of messages. Each message must have the right type. For example, a *query_if* message must be sent by an agent playing the "Train" role, like `treno_1`, to an agent playing the "Node" role, like `nodo_3`, and the arguments of the *query_if* content must contain the priority of the sender, the node from which the train will arrive and a coherent time interval. This message satisfies them:

```
msg(treno_1, nodo_3, query_if, free(2,240000,310000,nodo_1), cid(1), ts(1))
```

"Horizontal" properties of messages sequences. When a train contacts a sequence of nodes to verify whether they are free in order to optionally issue a reservation request, the arguments of the *query_if* messages must form a coherent path: the "From" argument in message m_{i+1} must be the same as the receiver of message m_i, the time slot's first extreme in message m_{i+1} must be the same as the time slot's second extreme in message m_i, the conversation id must be the same, and the train cannot change its priority, apart from setting it equal to infinity (`inf`) for requests than must necessarily be satisfied. For example, this trace (an extract of a real monitor log file) respects these constraints:

```
msg(treno_1,nodo_1,query_if,free(inf,156000,186000,init), cid(1),ts(1))
msg(treno_1,nodo_3,query_if,free(2,186000,256000,nodo_1), cid(1),ts(2))
msg(treno_1,nodo_4,query_if,free(2,256000,286000,nodo_3), cid(1),ts(3))
msg(treno_1,nodo_6,query_if,free(2,286000,326000,nodo_4), cid(1),ts(4))
```

Note that in the *query_if* message sent to the station entering node (in this case, `nodo_1`), the *From* field is set to the value *initial* (`init`) because there is no "coming from" node (the train is arriving from outside the station).

"Vertical" properties of conversations between a train and a node. Apart from the requirement that during a single conversation the train does not change the conversation id, we can identify one more constraint: if a node is reserved, it must inform the train that sent a $query\_if$ message of the arc it could have used to reach it and of the time slot when it will be free again. This time slot must start after the time slot's start indicated by the train in its $query\_if$ message, even if it may overlap with it. A trace like this (again from a real log file) respects both constraints:

```
msg(treno_1,nodo_3,query_if,free(2,24000,31000,nodo_1), cid(1),ts(1))
msg(nodo_3,treno_1,inform,reserved(da0,1,2,dummy,31001,38001),cid(1),ts(2))
```

Since a train can interact with the same node many times, for example because the attempt to reserve a path failed and then the train has to try to reserve a new one, we added and successfully tested another vertical constraint that involves conversation loops: if a train sends more than one $query\_if$ message to the same node, the conversation id must be different since the messages belong to different conversations.

Performances. Table 1 shows the performance analysis of three categories of execution: with our monitor, with the "plain" JADE Sniffer, with none of them.
– Column **Test** refers to the test we run among those discussed in Section 3.
– Column **R** (for **Runs**) reports the number of runs of a MAS. For example, R equal to 10 means that we performed 10 MAS executions with our monitor, 10 executions with the JADE Sniffer, and 10 executions with none.
– Column **Msg** (for **Messages**) reports the average number of messages exchanged among the agents per run. While in ABP and FYPA the average number is always the same as the exact number per run, as the MAS evolution is deterministic, in the ICNP MAS there is a random choice that participants can make about bidding or not. This means that the runs are not always the same and the number of messages per run can change. We run the MAS many times and we selected 5 runs for each execution category (with monitor, with JADE Sniffer, with none) which show homogeneous features, namely a number of iterations between the initiator and the participants between 4 and 7, and which guarantee that the average number of messages is the same for each category.
– Column **M** (for **Monitor**) reports the average number of milliseconds per message when using our monitor. This value changes from MAS to MAS, as deciding to send one message may require less or more reasoning from the agent, and hence less or more time. **JS** (for **JADE Sniffer**) reports the average number of milliseconds per message when using the JADE Sniffer and **N** (for **None**) reports the average number of milliseconds per message when using none of them.
– Column **M/JS (deg.)** reports the ratio between the performances with our monitor and with the JADE Sniffer and the degradation in percentage ("deg." field in round brackets). Similarly, **M/N (deg.)** reports the performances ratio and degradation between the execution with the monitor and with no JADE built-in agent, and **JS/N (deg.)** reports the performances ratio and degradation between the execution with the JADE Sniffer and with no JADE built-in agent.

For each test, we measured the complete execution time of the MAS. In particular, we measured the number of milliseconds between the start of the protocol (first

Table 1 Performances of the monitor execution

Test	R	Msg	M	JS	N	M/JS (deg.)	M/N (deg.)	JS/N (deg.)
ABP	10	20000	1.93	1.62	0.14	1.19 (19%)	13.78 (1278%)	11.38 (1038%)
ICNP	5	13	12.28	10.47	2.26	1.17 (17%)	5.43 (443%)	4.63 (363%)
FYPA1	5	12	8.10	8.05	2.77	1.01 (1%)	2.92 (192%)	2.90 (190%)
FYPA2	5	20	6.43	6.56	2.63	0.98 (-2%)	2.44 (144%)	2.49 (149%)
FYPA3	5	12	6.61	6.35	2.83	1.04 (4%)	2.33 (130%)	2.24 (124%)

message sent) and the protocol completion (last message received). Since the ABP is an infinite protocol, we measured the time between `bob`'s setup and the 10000th execution of its `action()` method.

In order to verify the portability of our framework across different operating systems, the experiments with FYPA were run on an Acer 7750 with Intel Core I5 2.3 GHz, 6 GB RAM and Windows 7 Home, whereas the others on an Acer TravelMate 6293 with Intel Core 2 Duo P8400/2.26 GHz, 4 GB RAM, and Mandriva Linux 2009 operating system.

Table 1 shows that the degradation due to the exploitation of the monitor agent with respect to the exploitation of the plain JADE Sniffer is usually between 1% and 19%, with only one test, FYPA2, where the monitor performed slightly better than the JADE Sniffer. The degradation when using the monitor should be mainly due to the fact that the monitor performs many I/O operations for writing the log both on file and on standard output. To make an example, the average dimension of the log files for our ABP tests is 300KB, which justifies the required additional time.

The JADE Sniffer agent is very time-consuming due both to its sniffing capabilities and to its complex graphical interface which requires updates on the fly. Using the JADE Sniffer w.r.t. not using it degrades the MAS performances up to 1038%. It is not surprising then the degradation due to the usage of the monitor w.r.t not using it, up to 1278%, since the monitor adds features to the JADE Sniffer.

From Table 1 we may also notice that the degradation of both the monitor and the JADE Sniffer with respect to using none worsens with the number of exchanged messages. In communication intensive MASs, the presence of agents like the JADE Sniffer and our monitor may represent a bottleneck. By implementing the monitor from scratch instead of relying on the Sniffer agent, keeping the textual interface and removing the GUI, by reducing the dimensions of the monitor's log files reporting only the identified problems, and by exploiting the projections presented in [2] that avoid bottlenecks due to the single centralized monitor, we are confident to overcome most problems related with the monitor's performance.

5 Related Work and Conclusions

Although there are many proposals for runtime verification of agent interaction protocols, that we carefully analyzed in our previous papers on this subject, the attempts

to integrate such mechanisms into JADE are, to the best of our knowledge, still missing.

Tools supporting the engineering of JADE MAS are described for example in [12] and [9]. In [12] data mining tools processing the results of the execution of large scale MASs in a monitored environment are discussed. They have been integrated in the INGENIAS Development Kit[8], in order to facilitate the verification of MAS models at the design level rather than at the programming level. The achieved results could be applied to JADE even if, to the best of our understanding, this has not been done. In [9], the authors present a unit testing approach for MASs based on the use of Mock Agents. Each Mock Agent is responsible for testing a single role of an agent under successful and exceptional scenarios. Aspect-oriented techniques are used to monitor and control the execution of asynchronous test cases. The approach has been implemented on top of JADE platform. None of these attempts has the same aim as ours, and thus those proposals and ours cannot be compared. Rather, they could be complemented for providing an integrated framework for engineering and developing JADE MASs.

The work probably most similar to ours, but not interfaced with JADE, is Scribble[9], a tool chain for runtime verification of distributed Java or Python programs against Scribble protocols specifications. Given a Scribble specification of a global protocol, the tool chain validates consistency properties and generates Scribble local protocol specifications for each participant (role) defined in the protocol. At runtime, an independent monitor is assigned to each Java (or Python) endpoint and verifies the local trace of communication actions executed during the session. Besides the different target languages, the main difference of Scribble w.r.t. our work is that we can monitor legacy MASs whose source code is not available because our monitor does not require any change to the agents' code, whereas the Scribble toolchain generates the executable code for the protocol endpoints starting from the specification of the protocol, hence it is suitable for monitoring systems which are created from the protocol specification, but not for legacy ones.

Our implementation of a JADE monitor agent suffers from some limitation, but our tests with three real MASs are very promising. The three problems that we experienced with our monitor are all related to the decision of extending the JADE Sniffer agent. The first is the one described in Section 2, regarding the messages order, the second arises when an agent that is under capturing by the monitor is born: the monitor needs some milliseconds to react and start capturing it, but if in the meanwhile the agent starts sending messages, the monitor could not receive them, and in this case a violation of the protocol is surely identified (even if it is a false positive). The last problem is related with performances, as discussed in Section 4.

We are studying a new version of the monitor that implements a JADE kernel service that captures all messages exchanged by the agents: in this way we should be able to avoid all the three problems above. A comparison with similar solutions including [5] and Scribble is also under way.

[8] `ingenias.sourceforge.net/`

[9] `http://www.scribble.org`

References

1. Ancona, D., Barbieri, M., Mascardi, V.: Constrained global types for dynamic checking of protocol conformance in multi-agent systems. In: SAC. ACM (2013)
2. Ancona, D., Briola, D., Seghrouchni, A.E.F., Mascardi, V., Taillibert, P.: Efficient verification of MASs with projections. In: EMAS Pre-proceedings (2014)
3. Ancona, D., Drossopoulou, S., Mascardi, V.: Automatic generation of self-monitoring MASs from multiparty global session types in Jason. In: Baldoni, M., Dennis, L., Mascardi, V., Vasconcelos, W. (eds.) DALT 2012. LNCS, vol. 7784, pp. 76–95. Springer, Heidelberg (2013)
4. Balachandran, B.M., Enkhsaikhan, M.: Developing multi-agent e-commerce applications with JADE. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) KES 2007, Part III. LNCS (LNAI), vol. 4694, pp. 941–949. Springer, Heidelberg (2007)
5. Baldoni, M., Baroglio, C., Capuzzimati, F.: 2COMM: A commitment-based MAS architecture. In: Winikoff, M. (ed.) EMAS 2013. LNCS, vol. 8245, pp. 38–57. Springer, Heidelberg (2013)
6. Briola, D., Mascardi, V.: Design and implementation of a NetLogo interface for the standalone FYPA system. In: WOA, pp. 41–50 (2011)
7. Briola, D., Mascardi, V., Martelli, M.: Intelligent agents that monitor, diagnose and solve problems: Two success stories of industry-university collaboration. J. of Inf. Assurance and Security 4, 106–117 (2009)
8. Briola, D., Mascardi, V., Martelli, M., Caccia, R., Milani, C.: Dynamic resource allocation in a MAS: A case study from the industry. In: WOA (2009)
9. Coelho, R., Kulesza, U., von Staa, A., Lucena, C.: Unit testing in multi-agent systems using mock agents and aspects. In: SELMAS, pp. 83–90. ACM (2006)
10. Mascardi, V., Ancona, D.: Attribute global types for dynamic checking of protocols in logic-based multiagent systems. TPLP 13(4-5-Online-Supplement) (2013)
11. Mascardi, V., Briola, D., Ancona, D.: On the expressiveness of attribute global types: The formalization of a real multiagent system protocol. In: Baldoni, M., Baroglio, C., Boella, G., Micalizio, R. (eds.) AI*IA 2013. LNCS, vol. 8249, pp. 300–311. Springer, Heidelberg (2013)
12. Serrano, E., Gómez-Sanz, J.J., Botía, J.A., Pavón, J.: Intelligent data analysis applied to debug complex software systems. Neurocomput. 72(13-15), 2785–2795 (2009)
13. Ughetti, M., Trucco, T., Gotta, D.: Development of agent-based, peer-to-peer mobile applications on ANDROID with JADE. In: UBICOMM (2008)

Part IV
Bio-Inspired Computing

A Genetic Approach for Virtual Computer Network Design

Igor Saenko and Igor Kotenko

Abstract. One of possible levels of computer protection may consist in splitting computer networks into logical chunks that are known as virtual computer networks or virtual subnets. The paper considers a novel approach to determine virtual subnets that is based on the given matrix of logic connectivity of computers. The paper shows that the problem considered is related to one of the forms of Boolean Matrix Factorization. It formulates the virtual subnet design task and proposes genetic algorithms as a means to solve it. Basic improvements proposed in the paper are using trivial solutions to generate an initial population, taking into account in the fitness function the criterion of minimum number of virtual subnets, and using columns of the connectivity matrix as genes of chromosomes. Experimental results show the proposed genetic algorithm has high effectiveness.

Keywords: data mining, genetic algorithms, VLAN, Boolean Matrix Factorization.

1 Introduction

Ensuring computer security is a very important task. The specific security threat may come from internal users (insiders). In many cases, the insiders' behavior prediction is a complex task. At the same time, a damage to computer security, which may be caused by unauthorized insiders' actions (malicious or deliberate) can be huge. Therefore, implementation of an additional level of computer security,

Igor Saenko
St.Petersburg Institute for Information and Automation of the Russian Academy of Sciences,
14-th line, 39, St.Petersburg, 199178, Russia
e-mail: ibsaen@comsec.spb.ru

Igor Kotenko
St.Petersburg Institute for Information and Automation of the Russian Academy of Sciences,
14-th line, 39, St.Petersburg, 199178, Russia
St. Petersburg National Research University of Information Technologies, Mechanics and Optics,
49, Kronverkskiy prospekt, St.Petersburg, Russia
e-mail: ivkote@comsec.spb.ru

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_11

designed to prevent unauthorized access of insiders to nodes of computer networks, is very desirable in all cases.

One of possible levels of computer protection against insiders is splitting computer networks into logical fragments that are known as *virtual computer networks* or, for simplicity, *virtual subnets*. This splitting is carried out through various means of VLAN (Virtual Local Area Network) technology [1]. These tools ensure that if two computers do not belong to the same virtual subnet, then the exchange of information between them is impossible. Using VLAN technology to differentiate information flows in the network should be very convenient for security administrators, because all actions necessary for that purpose can be localized in one place, for example, on an Ethernet switch or a router. An alternative way to differentiate information flows, which is associated with the use of access lists in operating systems for workstations in the computer network, is also possible, but it is less flexible. If it is needed to re-configure the information flow schema in this case, security administrators must access the remote workstations that may be not possible.

The task to create an additional level of computer protection against insider attacks based on virtual subnets is insufficiently discussed in the scientific literature. Usually, design of virtual subnets according to the VLAN technology is based on factors, not related to security. It is now largely accepted to form virtual subnets based on functional grounds. Virtual subnets are usually formed around servers giving access only permitted sets of computers. However, according to this approach the exchange of legitimate information between computers belonging to different subnets can be impossible. The paper investigates the possibility of building an additional level of security to ensure the required differentiation of information flows between computers based on the VLAN technology. The mathematical foundations of this problem are examined, showing that it is one of the varieties of Boolean Matrix Factorization (BMF). As the BMF methods can be considered as Data Mining methods and are applied to solve NP-complete problems, we believe this problem has the same computational complexity. The paper proposes to use genetic algorithms for solving this problem. Genetic algorithms have proved to be successful in solving optimization problems of different complexity. Some examples of their successful application in optimization problems, which deal with Boolean Matrixes decomposition, are known. However, the paper shows that it is not possible to apply directly known implementations of genetic algorithms to solve this problem.

The *main theoretical contribution* of the paper consists in the following: the first time we formulate the problem of virtual subnets design based on logical connectivity matrix; we show that this problem is one of the BMF forms; for solving the problem we propose an improved genetic algorithm. *The basic algorithm improvements* are as follows: using trivial solutions to generate an initial population; taking into account in the fitness function the criterion of minimum number of virtual subnets; using columns of the logical connectivity matrix as the genes of chromosomes.

The rest of the paper is structured as follows. Section 2 provides an overview of related work. Section 3 deals with mathematical foundations. The genetic algorithm to solve the task is described in section 4. Section 5 discusses the experimental results. Section 6 is a conclusion.

2 Related Work

In order to determine the scope of the problem and to form the mathematical basis for its decision, the works related to the decomposition of Boolean matrices were primarily analyzed. P. Miettinen et al. [2, 3, 4] considered some tasks of Matrix Factorization, which include Non-Negative Matrix Factorization (NMF) and BMF. They proved that NMF and BMF problems are NP-complete. Mathematical methods to solve them were suggested for these tasks. H. Lu et al. [5, 6] showed that some of the problems in the field of information security can be summarized by BMF. In particular, they demonstrated that the task of determining roles for role-based access model, known as Role Mining Problem (RMP), is one of these tasks.

These ideas were further elaborated in [7, 8], where the genetic algorithms to solve the RMP were proposed. In [7] a multi-chromosomal approach was suggested for coding solutions. According to this approach, each individual of genetic algorithm population has three chromosomes. One of these chromosomes is for controlling. It helps to perform crossover, when parent chromosomes have different length. In [8] a multi-chromosomal coding was denied and a stage of pre-processing was proposed for crossover. However, both these innovations could not be applied for designing virtual subnets, as the problem, which is discussed in the paper, requires determining only one Boolean matrix instead of two matrices. For this reason, the pre-processing stage, which is proposed in [8], should be considered as redundant. As it will be shown below, the problem does not require chromosomes with different lengths. Moreover, we propose an approach to calculate the length of chromosomes and to use so-called trivial solutions for generating initial population.

High complexity of the NMF and BMF problems and the ability to find effective mathematical methods to solve them only for special cases identified the need to find special heuristics. A. Janecek et al. [9] suggested using bio-inspired algorithms based on populations to meet the challenges of the NMF. They discussed several possible algorithms which can be used: genetic algorithms, particle swarm optimization, differential evolution, fish school search and fireworks algorithms between. Comparative evaluation of these algorithms showed that genetic algorithms have the greatest efficiency in solving NMF problems.

V. Snasel et al. [10, 11] proposed using genetic algorithms for solving BMF problem. These algorithms were designed to find the Boolean matrices $\mathbf{W}$ and $\mathbf{H}$, on which the given matrix $\mathbf{A}$ is decomposed. In this algorithm, rows of the matrix $\mathbf{H}$ are used as the genes of chromosomes and the crossover operator is applied to the columns of the matrix $\mathbf{W}$. The fitness function was formed on the base of Euclidean distance between the source and resulting matrices. In addition, this algorithm cannot strictly provide the equality between $\mathbf{W}$ and $\mathbf{H}^{\mathrm{T}}$, where $\mathbf{H}^{\mathrm{T}}$ is a transposed matrix $\mathbf{H}$. Due these features, the algorithm that was proposed in these works cannot be applied to our problem.

There is a few works on application of Data Mining to create virtual subnets. Ch.-F.Tai et al. [12] proposed to form a schema of virtual subnets by cluster analysis. This approach is oriented to use in mobile ad hoc networks. However, this approach is less efficient than genetic algorithms for large networks. The genetic algorithm for

optimization of virtual network schemes was proposed in [13]. But this algorithm allows forming only matrix $\mathbf{A}$ and does not allow finding matrices $\mathbf{X}$ and $\mathbf{X}^{\mathrm{T}}$, on which matrix $\mathbf{A}$ should be decomposed. In addition, this approach does not take into account the criterion of minimizing the total number of virtual subnets.

Thus, the analysis of relevant work has shown that genetic algorithms should be considered as sufficiently effective to solve the problem discussed. At the same time, genetic algorithms, which are proposed in known works, cannot be directly used for this purpose.

3 Mathematical Background

Suppose that there are n computers in a network. The diagram of allowed flows between these computers is defined by Boolean matrix $\mathbf{A}[n, n]$. If $a_{ij} = 1(i, j = 1, \ldots, n)$ then the exchange between computers i and j is allowed. Otherwise, this exchange is not possible.

Suppose that we have formed k virtual subnets in the network. Each of these subnets combines two or more computers. Let us set the distribution of computers on subnets by using the matrix $\mathbf{X}[n, k]$. If $x_{ij} = 1(j = 1, \ldots, k)$ then the computer i belongs to the subnet j. Otherwise subnet j does not include computer i.

The matrix $\mathbf{A}$ plays a role of initial data. The matrix $\mathbf{X}$ is the result of solving the problem. We will show that between Boolean matrices $\mathbf{A}$ and $\mathbf{X}$ there is the following relationship:

$$\mathbf{A} = \mathbf{X} \otimes \mathbf{X}^{\mathrm{T}} \tag{1}$$

where $\mathbf{X}^{\mathrm{T}}$ is a transposed matrix $\mathbf{X}$, symbol $\otimes$ stands for Boolean matrix multiplication, which is a form of matrix multiplication based on the rules of Boolean algebra. Boolean matrix multiplication allows getting the elements of matrix $\mathbf{A}$ by the following expression: $a_{ij} = \vee_{j=1}^{n}(x_{ij} \wedge x_{ji})$.

Consider the following example. Let $n = 5, k = 3$. There are virtual subnets in the computer network as shown in Fig. 1. Subnet 1 includes workstations WS1, WS2 and WS4, subnet 2 — WS2, WS3 and WS5, subnet 3 — WS1, WS4 and WS5.

In this case, the relationships between the matrices $\mathbf{A}$ and $\mathbf{X}$ are as follows:

$$\mathbf{X} = \begin{pmatrix} 1\,0\,1 \\ 1\,1\,0 \\ 0\,1\,0 \\ 1\,0\,1 \\ 0\,1\,1 \end{pmatrix}, \mathbf{X}^{\mathrm{T}} = \begin{pmatrix} 1\,1\,0\,1\,0 \\ 0\,1\,1\,0\,1 \\ 1\,0\,0\,1\,1 \end{pmatrix}, \mathbf{A} = \begin{pmatrix} 1\,1\,0\,1\,1 \\ 1\,1\,1\,1\,1 \\ 0\,1\,1\,0\,1 \\ 1\,1\,0\,1\,1 \\ 1\,1\,1\,1\,1 \end{pmatrix}.$$

It is easy to see that the matrix $\mathbf{A}$ is symmetric.

Now we will show that this problem is a kind of BMF problems. The BMF problem is to find Boolean matrices $\mathbf{W}$ and $\mathbf{H}$ that are related with given Boolean matrix $\mathbf{A}$ by the following equation:

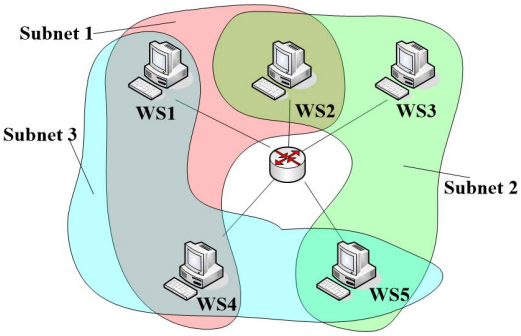


Fig. 1 Distribution of workstations over virtual subnets (example)

$$\mathbf{A} = \mathbf{W} \otimes \mathbf{H}, \tag{2}$$

where $\mathbf{A} = \mathbf{A}[n, m]$, $\mathbf{W} = \mathbf{W}[n, k]$ and $\mathbf{H} = \mathbf{H}[k, m]$.

Comparing (1) and (2), we can notice that the equation (1) can be seen as a special case of the equation (2) when the next two conditions are met. The first condition is the equality $m = n$. The second condition takes the following form:

$$w_{ij} = h_{ji} \text{ for any } i = 1, \ldots, n \text{ and } j = 1, \ldots, k. \tag{3}$$

From the fact that the problem discussed is a variant of BMF problems, it follows that the problem is NP-complete. However, because of conditions (3), the direct application of the BMF methods for solving the problem is difficult. Hence our proposal is to solve this problem by heuristic methods, namely, genetic algorithms.

However, for this purpose, the optimization criterion should be defined. Let us pay attention to the fact that in the task (1) the value k can be arbitrary. Now we show how we can restrict it from above. For this purpose we will split the network on subnets, where each subnet includes only two computers i and j when $a_{ij} = 1$.

For the example shown in Fig. 1 the matrix $\mathbf{X}$ has five rows and eight columns:

$$\mathbf{X} = \begin{pmatrix} 1\,1\,1\,0\,0\,0\,0\,0 \\ 1\,0\,0\,1\,1\,1\,0\,0 \\ 0\,0\,0\,1\,0\,0\,1\,0 \\ 0\,1\,0\,0\,1\,0\,0\,1 \\ 0\,0\,1\,0\,0\,1\,1\,1 \end{pmatrix}.$$

The matrix $\mathbf{X}^{\mathrm{T}}$, respectively, has eight rows and five columns. However, Boolean multiplication of these matrices still results in a matrix $\mathbf{A}$ (it is not hard to check). The order of the columns can be arbitrary.

We will call such a matrix $\mathbf{X}$ as *the trivial solution*. A trivial solution has the following properties. First, the number of columns in the matrix $\mathbf{X}$ is equal to the number of '1'-values in the matrix $\mathbf{A}$ located above the main diagonal.

We denote this value as M. Secondly, in each column of the matrix $\mathbf{X}$ there is just two '1'-values. Finally, a number of trivial solutions is equal to the number P of permutations of columns in the matrix $\mathbf{X}$. It is obvious that $P = 2^M$. For this reason the trivial solutions cannot be considered as good. At the same time, we see that for the example shown in Fig. 1 there is a good solution, in which $k = 3$. Therefore, we propose the *optimization criterion* of the problem as the *minimum value of k under full coincidence* of matrices $\mathbf{A}$ and $\mathbf{X} \otimes \mathbf{X}^{\mathrm{T}}$. The smaller k, the less the complexity of administrative work and higher the network security.

4 Genetic Algorithm

Genetic algorithms are well known method of bio-inspired optimization. However, there are different views on the sequence and content of the steps of genetic algorithms [14, 15, 16]. The paper proposes to follow the following sequence of steps:

1. *Defining the fitness function,* which shows why one solution better than the other solution.

2. *Encoding possible solutions of the problem.* The encoded solution in the terminology of genetic algorithms is called *individual*. Usually solutions are encoded using character or numeric strings. A single character of this code is called *gene*. A set of genes in a string is called *chromosome*.

3. *Creating the initial set of individuals* that called *population*. Typically, this process takes place at random, but the number of individuals in the population N is permanent. This step includes also evaluating all individuals in a population through fitness function and sorting them in descending order of its value.

4. Choosing pairs of individuals, called *parents*, to form new individuals, called *descendants*, by *crossing*. Parents' chromosomes are broken into fragments. Then the fragments of parental chromosomes are crossed to form descendant chromosomes. New individuals are assessed using fitness function and are added to the current population.

5. Choice of individuals for *mutations*, the mutation and evaluation of these individuals. Under mutation, the individuals modify their genes.

6. *Selection of the population*, which consists in reservation of the N individuals possessing the highest values of fitness functions. Other individuals (the "bad") are removed from the current population.

7. If the algorithm termination criteria are fulfilled, then an individual with a maximum value of the fitness function is a solution. Otherwise, return to step 3.

In order to use the genetic algorithm to solve the problem discussed, it is necessary to determine the fitness function and coding decisions.

In the known papers on the application of genetic algorithms for solving the BMF problem, the fitness function was based on the Euclidean distance between the matrices $\mathbf{A}$ and $\mathbf{W} \otimes \mathbf{H}$. Under full matching of these matrices the fitness function has the maximum value. In our case only one Euclidean distance between $\mathbf{A}$ and $\mathbf{X} \otimes \mathbf{X}^{\mathrm{T}}$ is not sufficient. The trivial solutions may be obtained from this one

criterion. We must also take into account the requirement that in the matrix $\mathbf{X}$ the number of columns k should be minimal. So we propose the following type of fitness functions:

$$F = \left(\alpha k + \beta \sqrt{\sum_i \sum_j (a_{ij} - x_{ij} x_{ji})^2} \right)^{-1}, \tag{4}$$

where α and β are the weights that determine the direction of the solution search. The condition $\alpha \ll \beta$ between these coefficients guarantees that first of all we want to look for solutions which matrices $\mathbf{A}$ and $\mathbf{X} \otimes \mathbf{X}^{\mathrm{T}}$ are the same, then we will look for solutions with smaller values of k.

Now, let us consider the order to create chromosomes. To do this, we propose to use as genes of chromosomes the vector $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in})$. The vector $\mathbf{x}_i$ plays the role of a column of the matrix $\mathbf{X}$. Let us set the number of genes in the chromosome equal M. In fact, the number M specifies the number of columns in a trivial solution. More columns make no sense.

Finally, in order to increase the convergence of the genetic algorithm we will make another proposal, which deals with the formation of the initial population. We will form it from possible trivial decisions. If we get $P \leq N$, then the initial population will consist of trivial solutions. Otherwise P individuals are trivial solutions, and the other $(N - P)$ individuals will be randomly generated.

Operation of crossing and mutation will be performed in a standard way.

5 Experimental Results

The proposed approach to the formation of an additional level of protection for the computer network was implemented in a computer network that was physically deployed on three floors of a building. It used Cisco Catalyst 5000 Series switches. The dynamic VLAN technology was implemented based on MAC addresses. For each computer the access lists to the neighboring computers both at the level of one floor of the building and between floors was made by the system administrator. Access lists have served as the basis for the formation of the original matrix $\mathbf{A}$. Each access list forms one row in $\mathbf{A}$. The total number of computers covered under the VLAN and, accordingly, specified by the matrix $\mathbf{A}$, is equal to 100. The larger number of computers was considered inappropriate because of difficulties in forming the matrix $\mathbf{A}$. Using the genetic algorithm, a solution was found, that consists in splitting the network to about 50 of large and small virtual subnets, which were formed by the security administrator. Users have not identified the decrease of the network performance. However, the users were completely prevented from making unauthorized access to neighboring computers.

For evaluation of speed and accuracy of the genetic algorithm, the testbed was developed, the structure of which is shown in Fig. 2.

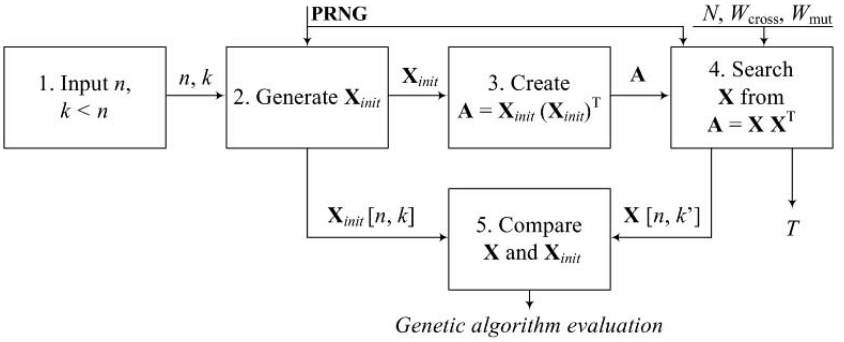


Fig. 2 Testbed for the genetic algorithm assessment

The parameters n and $k(k < n)$ are formed in the module 1 as the *initial data of the testbed*. Using a pseudorandom number generator (PRNG) [17], the initial matrix $\mathbf{X}_{\text{init}}$ of computer distribution over virtual subnets is randomly generated in the module 2. In the module 3, the matrix $\mathbf{A}$ is calculated as the initial data of the problem based on the formula $\mathbf{A} = \mathbf{X}_{\text{init}} \otimes (\mathbf{X}_{\text{init}})^{\text{T}}$. Then, in the module 4, taking into account the matrix $\mathbf{A}$ and using the genetic algorithm, the matrix $\mathbf{X}[n, k']$ is determined, which is the solution of the problem. In general case, the number of columns k' in the matrix $\mathbf{X}$ is not equal to the value of k. If $k' \leq k$, then we consider that the algorithm has found the exact solution of the problem, otherwise — approximate. The algorithm parameters are: number of individuals in the population (N), the probability of choosing the individuals for crossover operation (W_{cross}) and the probability of choosing the individuals for the mutation (W_{mut}). These values were taken from [10]: $N = 200, W_{\text{cross}} = 0.1, W_{\text{mut}} = 0.01$. The outputs of the module 4 are matrix $\mathbf{X}$ and the number of iterations of the algorithm (T). The maximum number of iterations was set to $T_{\max} = 1000$. The exceeding of this value leads to the algorithm completion. Finally, the module 5 fulfills the final evaluation of the algorithm by comparing matrices $\mathbf{X}$ and $\mathbf{X}_{\text{init}}$.

The speed assessment for the algorithm was evaluated, while the number of iterations did not reach the limit (i.e. $T < T_{\max}$). The measure of speed was the average of the number of iterations $(\bar{T})$, determined for the pair of values (n, k). Five tests were carried out for each pair (n, k). The n value were changed in the range from 10 to 100. The k value had the following values: $0.2n, 0.3n$ and $0.5n$.

The accuracy δ of the algorithm was evaluated when the iteration count reached the limit (i.e. $T = T_{\max}$) according to the following formula:

$$\delta = [(n - \max(0; k' - k))/n] \cdot 100\%. \tag{5}$$

In this case, if the condition $k' = k$ hold true, then regardless of the n value the accuracy is 100%. If $k' > k$, then the accuracy depends on the relationship between the $(k' - k)$ and the n values. So, if $n = 50, k = 10$ and $k' = 11$, then $\delta = 98$.

The results of the speed and accuracy evaluation for the genetic algorithm proposed in the paper are outlined in Tab. 1.

Table 1 Experimental results

n	k	$\bar{T}$	$\delta, \%$	n	k	$\bar{T}$	$\delta, \%$
10	2	48.3	100	50	25	534.6	100
10	3	35.6	100	75	15	1000.0	91
10	5	25.8	100	75	23	1000.0	95
25	5	255.8	100	75	38	1000.0	97
25	8	183.7	100	100	20	1000.0	78
25	13	127.5	100	100	30	1000.0	85
50	10	1000.0	98	100	50	1000.0	90
50	15	811.4	100				

Analyzing the data presented in Tab. 1, it is possible to draw the following conclusions. First, under the small dimensions of the problem ($n = 10$ and $n = 25$) the proposed algorithm allowed to generate the solution with maximum accuracy δ = 100% in real or near-real-time. Secondly, for large dimensions ($n = 75$ and $n = 100$) it was impossible to obtain the maximum accuracy in the allotted time. However, the accuracy is quite large and ranges from 78 to 97 percent. Finally, when n is constant, the complexity of solving the problem increases when k decreases. It is well illustrated by the $n = 50$. When $k = 15$ and $k = 25$, the most accurate solutions were received; and when $k = 10$, the accuracy of the solution was 98 percent.

The analysis of experimental results leads to the conclusion that the proposed algorithm has sufficiently high performance for designing virtual computer networks.

6 Conclusion

The paper proposed the approach to the construction of an additional level of protection for computer network based on the formation of virtual subnets according to the specified requirements to logical connectivity between computers. As the main tool to create virtual subnets, VLAN technology is considered.

Analysis of the mathematical statement of the problem showed that it is a type of Boolean Matrix Factorization. However, as distinct from the BMF which looks for two different Boolean matrices, the considered problem searches only one Boolean matrix, such that the multiplication of this matrix and its transposed image gives the desired result. For this reason, it was concluded that using known mathematical

methods of the BMF directly is inappropriate. To solve the problem, the paper suggests to use genetic algorithms. It was shown that known genetic algorithms used for solving the problems of BMF or virtual network schema optimizations cannot be directly used and require some improvements. Key improvements proposed in the paper are: the use of trivial solutions to generate an initial population; taking into account in the fitness function the criterion of minimum number of virtual subnets; the use columns of the connectivity matrix as genes of the chromosomes encoding the solutions. Experimental evaluation of speed and accuracy for the proposed genetic algorithm has shown its high efficiency. Future studies are directed to the application of the proposed approach to the reconfiguration of virtual subnets.

Acknowledgements. This research is being supported by the grants of the Russian Foundation of Basic Research (13-01-00843, 13-07-13159, 14-07-00697, 14-07- 00417), the Program of fundamental research of the Department for Nanotechnologies and Informational Technologies of the RAS (contract #2.2), and by Government of the Russian Federation, Grant 074-U01, and State contract #14.604.21.0033.

References

1. Catalyst 2900 Series XL and Catalyst 3500 Series XL Software Configuration Guide. Cisco IOS Release 12.0(5) WC(1). Cisco Systems, San Jose (2001)
2. Miettinen, P., Vreeken, J.: Model Order Selection for Boolean Matrix Factorization. In: 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. ACM, New York (2011)
3. Miettinen, P.: Dynamic Boolean Matrix Factorizations. In: 2012 IEEE 12th International Conference on Data Mining. ACM, New York (2012)
4. Cergani, E., Miettinen, P.: Discovering Relations using Matrix Factorization Methods. In: 22nd ACM International Conference on Information & Knowledge Management. ACM, New York (2013)
5. Lu, H., Vaidya, J., Atluri, V., Hong, Y.: Extended Boolean Matrix Decomposition. In: Ninth IEEE International Conference on Data Mining. IEEE Press, New York (2009)
6. Lu, H., Vaidya, J., Atluri, V.: Optimal Boolean Matrix Decomposition: Application to Role Engineering. In: 24th IEEE International Conference on Data Engineering. IEEE Press, New York (2008)
7. Saenko, I., Kotenko, I.: Genetic Algorithms for Role Mining Problem. In: 19th International Euromicro Conference on Parallel, Distributed and Network-Based Processing. IEEE Press, New York (2011)
8. Saenko, I., Kotenko, I.: Design and Performance Evaluation of Improved Genetic Algorithm for Role Mining Problem. In: 20th International Euromicro Conference on Parallel, Distributed and Network-based Processing. IEEE Press, New York (2012)
9. Janecek, A., Tan, Y.: Using Population Based Algorithms for Initializing Nonnegative Matrix Factorization. In: Tan, Y., Shi, Y., Chai, Y., Wang, G. (eds.) ICSI 2011, Part II. LNCS, vol. 6729, pp. 307–316. Springer, Heidelberg (2011)
10. Snasel, V., Platos, J., Kromer, P.: On Genetic Algorithms for Boolean Matrix Factorization. In: Eighth International Conference on Intelligent Systems Design and Applications, vol. 2, pp. 170–175. IEEE Press, New York (2008)
11. Snasel, V., Platos, J., Kromer, P., Husek, D., Neruda, R., Frolov, A.A.: Investigating Boolean Matrix Factorization. In: Workshop on Data Mining using Matrices and Tensors (2008)
12. Tai, C.-F., Chiang, T.-C., Hou, T.-W.: A Virtual Subnet Scheme on Clustering Algorithms for Mobile Ad Hoc Networks. Expert Systems with Applications 38(3), 2099–2109 (2011)

13. Saenko, I., Kotenko, I.: Genetic Optimization of Access Control Schemes in Virtual Local Area Networks. In: Kotenko, I., Skormin, V. (eds.) MMM-ACNS 2010. LNCS, vol. 6258, pp. 209–216. Springer, Heidelberg (2010)
14. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Longman Publishing, Boston (1989)
15. Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Massachusetts (1998)
16. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer (2007)
17. Barker, E., Kelsey, J.: Recommendation for Random Number Generation Using Deterministic Random Bit Generators. NIST Special Publication. NIST (2012)

Gene Expression Programming for Evolving Two-Dimensional Cellular Automata in a Distributed Environment

César Manuel Vargas Benítez, Wagner Weinert, and Heitor Silvério Lopes

Abstract. This paper presents a novel distributed bio-inspired approach that uses Gene Expression Programming (GEP) to evolve transition rules for two-dimensional Cellular Automata (2D-CA). The 2D-CA are simulated in parallel using a master-slave distributed environment. The fitness function of the GEP ultimately measures the ability of a given CA to create a suitable solution for a complex Bioinformatics problem. To validate the proposed approach, extensive experiments were done dealing with a computationally expensive problem, that is considered to be one of the most important open challenges in Bioinformatics. Results of simulations show that the proposed approach was effective for the problem. Future works will investigate other distributed approaches of this approach, such as those based on General-Purpose Graphics Processing Units (GPGPU) or hardware-based accelerators. Finally, we believe that the method proposed in this work can be useful for other computational problems.

Keywords: Bio-inspired Computing, Distributed Computing, Gene-Expression Programming, Cellular Automata, Self-organization, Emergent Behavior, Contact Maps, Protein Folding Problem.

1 Introduction

Stephen Wolfram proposed a "New kind of Science" that is based on general types of rules that can be embodied in simple computer programs for reproducing

César Manuel Vargas Benítez · Heitor Silvério Lopes
Bioinformatics Laboratory, Federal University of Technology - Paraná (UTFPR), Curitiba, Brazil
e-mail: {cesarbenitez,hslopes}@utfpr.edu.br

Wagner Weinert
Federal Institute of Education Science and Technology of Paraná (IFPR), Paranaguá, Brazil
e-mail: wrweinert@gmail.com

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_12

real-world complex behaviors, instead using traditional mathematical methods [17]. A particular class of such computer program are the Cellular Automata (CAs), which are simple discrete idealizations of natural systems. CAs are families of simple, finite-state machines that exhibit emergent behaviors through their interactions [8].

The computational simulation of a CA system is relatively simple, where a configurational state of the CA is determined according to its predecessor state and a transition rule. However, finding a transition rule for a given dynamic behavior is a very difficult task, for which there is no efficient method [16].

The main objective of this work is to propose a novel parallel approach for the induction of transition rules of two-dimensional Cellular Automata (2D-CA), using Gene Expression Programming, for solving complex problems in a reasonable computing time. The second objetive, not less important, is to apply the proposed approach to the Protein Contact Map prediction, validating the proposed approach and proposing a novel method for the Protein Folding Prediction Problem (PFP).

This paper is organized as follows: Section 2 presents an overview about Cellular Automata; Section 3 describes the Gene Expression Programming (GEP); Section 4 presents an overview about the PFP; Section 5 describes the proposed approach; Section 6 shows how the experiments were conducted and the results; finally, in Section 6 conclusions and future works are presented.

2 Cellular Automata (CA)

Cellular Automata (CAs) were introduced by John von Neumann in his work on *self-reproducing machines* [12] and have been used to model several biological, physical and engineering systems. For instance, simulation of the HIV infection dynamics [19], and water flow simulation [14]. Basically, CAs are discrete dynamic systems that are be represented by a d-dimensional array, composed of identical interconnected components (cells).

The dynamic behavior of a CA is represented by its spatio-temporal diagram [17]. Each cell has a discrete state that is updated on discrete time steps, considering its current state and the state of the neighboring cells (neighborhood relationship). All cells of the d-dimensional matrix are updated at the same time step by the application of a transition rule. Thus, sucessive applications of the transition rule will lead to a dynamic behavior, from the initial state in t_0 to successive states in subsequent time steps $(t_1, \cdots, t_n)$.

The following formal notation for CAs is presented by [11]: $\sum$ set of possible states of a cell; k number of elements of the set $\sum$; i index of a specific cell; S_i^t state of a cell in a given time t; η_i^t neighborhood of cell i; $\Phi(\eta_i^t)$] transition rule that defines the next state S_i^{t+1} for each cell i, as function of η_i^t.

The neighborhood of each cell $(c_{i,j})$ of a two-dimensional Cellular Automaton (2D-CA) is composed following a neighborhood relationship. The neighborhood relationship is determined by a predefined radius (r) and the size (number of cells) of the neighborhood (m) is defined as a function of r, acoording to $m = 2r + 1$.

The most common types of neighborhood for 2D-CA are the von Neumann and the Moore neighborhoods. The size of the von Neumann neighborhood with $r = 1$ is $m = 5$, comprising the four orthogonally neighboring cells ($c_{i-1,j}$, $c_{i,j-1}$, $c_{i+1,j}$, $c_{i,j+1}$) surrounding a central cell ($c_{i,j}$). On the other hand, the Moore neighborhood is composed by the central cell and eight neighboring cells ($c_{i-1,j}$, $c_{i,j-1}$, $c_{i+1,j}$, $c_{i,j+1}$, $c_{i-1,j-1}$, $c_{i+1,j-1}$, $c_{i-1,j+1}$, $c_{i+1,j+1}$).

In addition, boundary conditions are used to allow the connection between cells that are situated at the extremities, forming a toroidal arrangement. Thus, the transition rule ($\Phi(\eta_i^t)$) is applied over all cells of the CA, without failure.

The number of transitions (possible configurations of the neighborhood) that compose the rule Φ is given by k^{2r+1} and the number of rules represented by those transitions is $k^{k^{2r+1}}$. For instance, the rule of a binary 2D-CA with von Neumann neighborhood (with $r = 1$) is composed of 32 transitions. In other words, the rule is composed of 32 bits, where each bit represents the result of a transition (i.e. $c_{i,j}^{(t)} \rightarrow c_{i,j}^{(t+1)}$), according to the concept of elementary automata proposed by Wolfram.

3 Gene-Expression Programming (GEP)

Gene-Expression Programming (GEP) is an extension of Genetic Programming (GP) that was proposed by [5]. The difference between these approaches lies in how the individuals are represented. In GP, the individuals are nonlinear entities of different sizes and shapes (concrete syntax trees). On the other hand, in GEP the individuals are encoded as fixed-size linear strings (also known as genome or chromosome), which are afterwards expressed as nonlinear entities of different sizes and shapes (i.e. expression trees or diagram representations) [5].

The encoding of individuals in GEP is based on the biological concept of open read frame (ORF), that is the coding sequence of a gene. However, it is important to know that genes are composed of more sequences than the respective ORF. In biology, an ORF is composed of amino acid codons, beginning with a "start" codon and ending at a termination codon. GEP genes are composed of a head and a tail. The head has symbols that represent functions and terminals. On the other hand, the tail contains only terminals. GEP genes can have noncoding regions, that, in fact, are the essence of GEP, allowing modifications of the genome using any genetic operator without restrictions, producing valid programs without the need for editing processes [5]. In other words, the encoding region of a gene (ORF) can "activate" or "deactivate" portions of the genetic material algorithm, according to the functions and their arities (i.e. number of arguments) encoded in the head of the gene.

GEPCLASS [18] is an implementation of GEP specially designed for finding rules for classification problems based on supervised learning, where it is aimed to find rules for modeling a given domain of known data samples, and then classify unseen sets of data. In GEPCLASS, a population of individuals evolves for a number of generations, where selected individuals are subjected to genetic operators (mutation,

recombination and transposition), generating diversity and, consequently, allowing the evolutionary process to continue for more generations, increasing the chances of finding even better solutions. The head of the gene can have elements belonging only to the set of functions, such as logical and comparison operators.The tail, in turn, can have elements either from the set of functions or from the set of terminals, which in turn, includes the attributes that describe particular values.

The mapping between the genotype to the phenotype is carried out as follows. The chromosome is transcribed into a variable-size expression tree (ET), following the Karva language [5], where each gene is trascribed to a separated sub-tree. Then, all sub-trees are joined together by a linking function (*AND* or *OR* operator), composing the ET that represents a candidate solution to a given problem. The quality of the candidate solutions is measure by a fitness function.

4 The Protein Folding Problem and the Protein Contact Maps

Under physiological conditions every protein folds into a unique three-dimensional structure, also known as the native tertiary structure or native conformation, that determines their specific biological function. This process is known as the Protein Folding.

Despite the considerable theoretical and experimental effort expended to study the protein folding process, there is not yet a detailed description of the mechanisms that govern the folding process.

Although the concept of the folding process arose in the field of Molecular Biology, the Protein Folding Problem (PFP) is clearly interdisciplinary, requiring support of many knowledge areas, and it is considered to be one of the most important open challenges in Biology and Bioinformatics.

Better understanding the protein folding process could help to: (a) accelerate drug discovery by replacing slow, expensive structural biology experiments with faster computational simulations, and (b) infer protein function from genome sequences.

Contact Maps (CM) are minimalistic representations of protein structures. The contact map for a protein sequence with N amino acids is a $N \times N$ binary symmetrical matrix (C), which is defined as follows: $C_{i,j} = 1$ if residues ith and jth are in contact, otherwise $C_{i,j} = 0$. Each position of the matrix (ith,jth) is 1 if the amino acid pair (ith and jth amino acids) fulfills the connectivity condition. One can define a contact between two amino acids in different ways. For instance, we can consider two amino acids in contact when their Cα atoms are closer than a arbitrary threshold distance [4].

As commented in last section, the solution of the folding problem is still lacking. Among different possibilities, the prediction of protein contact maps is particularly promising, since even a partial solution of it can significantly help the prediction of the protein structure [4]. Several methods have been developed for CM prediction from sequence. For instance, Neural Networks (NN) [7], Genetic Programming (GP) [9] and neuro-fuzzy systems [1].

5 Implementation of the Parallel GEP-CA (pGEP-CA)

Algorithm 2 shows the pseudo-code of the pGEP-CA. A parallel master-slave architecture is employed in order to allow a reasonable computing time. The processing load is divided into several processors (slaves), under the coordination of a master processor. The master is responsible for initializing the population, determining the ORFs, performing the selection procedure, applying the operators(clone operator, mutation, recombination, IS (insertion sequence) transposition, RIS (root IS) transposition and genic transposition) based on the GEPCLASS [18] and distributing individuals to the slaves. Slaves, in turn, are responsible for reading the initial and final 2D-CAs, simulating the 2D-CAs from the initial 2D-CA, using the induced rules and computing the fitness function of each individual received, using the final (expected) 2D-CAs and the obtained 2D-CAs. In Algorithm 2, bold instructions are processed in parallel. The software was developed in ANSI-C programming language, using the Message Passing Interface (MPI) for communication between processes[1] and the Mersenne Twister random number generator [10].

Algorithm 2. Pseudo-code for parallel GEP-CA (pGEP-CA)

```
1: Start
2: Initialize population;
3: Determine ORFs
4: Simulate 2D-CAs and Evaluate fitness in parallel
5: while stop criteria not satisfied do
6:     Clone operator – part 1
7:     Selection
8:     Apply genetic operators
9:     Update population
10:    Determine ORFs
11:    Simulate 2D-CAs and Evaluate fitness in parallel
12:    Clone operator – part 2
13: end while
14: Export postprocess results: best transition rule, obtained CM, metrics
15: End
```

5.1 Solution Encoding and Fitness Function

First of all, it is important to know how the transition rule is composed. As commented in Section 2, the transition rule is formed by concatenating all transitions, which in turn, are defined by the possible combinations of the neighboring cells. The number of combinations, using the von Neumann neighborhood with unity radius, is 32, obtained as shown in Section 2. For instance, the rule

[1] Available at: `http://www.mcs.anl.gov/research/projects/mpich2/`

"$01010111011111110111111101100101_2$" is formed by the transitions, from right to left, "00000_2"→1; "00001_2"→0; "00010_2"→1; $\cdots$; "11111_2"→0.

In this work, the encoding of the individuals is defined according to the set of terminals and their domains, following the Pittsburgh approach [6]. The terminals are binary and represent the state of the neighboring cells ('1'=contact, '0'=non-contact). The set of terminals represent all possible combinations of the neighborhood, mapping all transitions of a given rule. Considering the von Neumann neighborhood with $r = 1$, the terminal set is composed of five terminals (labeled as a, b, c, d and e). The terminals have the same domain, which in turn, is defined by the possible states of the cells of the CA ($\sum = [0; 1]$).

The individuals are represented by two multigenic chromosomes, which in turn, are composed of more than one gene of equal size. As stated in Section 3, every gene is divided into a head and a tail. The size of the head (h) of each gene and the number of genes of each chromosome can be chosen *a priori*. On the other hand, the size of the tail (t) is determined according to the size of the head as proposed by [18]: $t = \text{IntegerPart}[0.5(h(n-1)+1)]$, where n represents the largest arity (number of arguments) of the functions used.

Each gene is directly translated into an expression tree (ET). In this work, each chromosome is composed of two genes. Thus, the sub-ETs codified by the genes are linked together by a logical function (*AND* or *OR*), which can be chosen *a priori*.

A possible transition encoded in an individual is written in the form *IF* A *THEN* C as in data classification systems. For example, a possible rule encoded in an individual would be: *IF* ($a = 1$ *AND* $b = 0$) *THEN* $Rule$ = *'1'; else* $Rule$ = *'0'*.

Figure 1 shows a simplified example of the transcription process.

Contact maps (CMs) are generally sparse symmetric matrices, populated primarily with non-contacts (or zeros). Therefore, a similarity measure between two CMs based on the Hamming distance [13] or Euclidean distance does not work well for CMs, because contacts (true) and non-contacts (false) values carry the same weight. Therefore, the fitness function proposed in this work is based on three metrics, that are better suited to this problem, chosen to measure the ability of a transition rule to generate a CA that represents a CM correctly. The fitness function is shown in Equation 1.

$$fitness = S_C * S_{NC} * S_i^2 \tag{1}$$

where: S_C, S_{NC} are based on the sensitivity and specificity measures, respectively. Sensitivity and specificity are commonly used in classification systems. Sensitivity measures the ability of the classifier to correctly assign a data to its real class. On the other hand, specificity measures the ability to reject a given data as belonging to a class to which it does not belong. In this work, S_C and S_{NC} measure the ability of a transition rule to generate correct contacts and non-contacts, respectively. S_C and S_{NC} are defined following four types of result, as shown in Equations 2 and 3, respectively. S_i measures the symmetry of the CM, as shown in Equation 4, where m and n are the number of rows and columns of the CM, respectively.

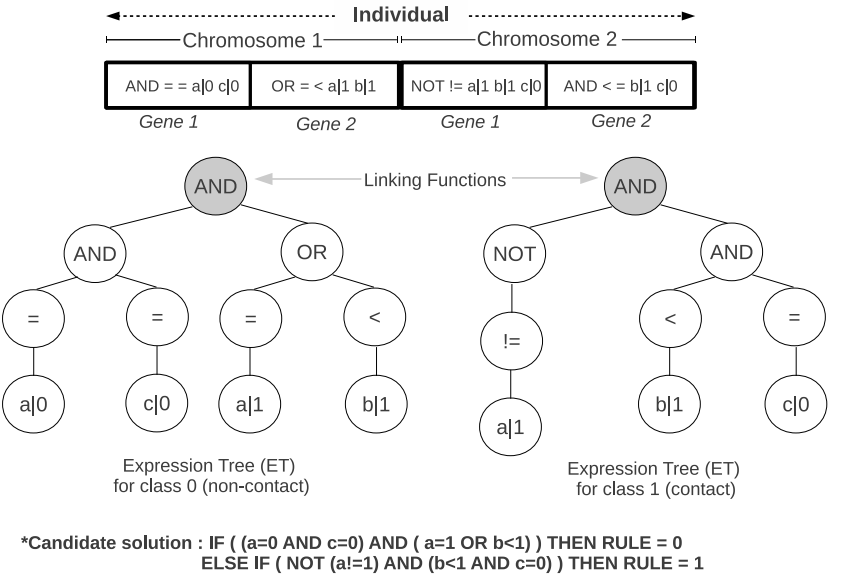


Fig. 1 GEP Transcription process – example

$$S_C = \frac{T_C}{T_C + F_{NC}} \quad (2) \qquad S_{NC} = \frac{T_{NC}}{F_C + T_{NC}} \quad (3) \qquad S_i = \sum_{i=0}^{i<m-1} \sum_{j=i+1}^{j<n} [1 - |(C_{i,j} - C_{j,i})|] \quad (4)$$

where:

- **True contacts** (T_C)**:** number of contacts generated by the transition rule that, in fact, are contacts;
- **True non-contacts** (T_{NC})**:** number of non-contacts generated by the transition rule that, in fact, are non-contacts;
- **False contacts** (F_C)**:** it counts the contacts generated by the transition rule that, in fact, are non-contacts;
- **False non-contacts** (F_{NC})**:** it counts the non-contacts generated by the transition rule that, in fact, are contacts;
- $C_{i,j}$ represents the value at cell location (i, j) of the CM.

6 Computational Experiments and Results

All experiments done in this work were run in a cluster of networked computers. Each computer has an Intel Core-2 Quad processor at 2.8GHz. All computers run a minimal installation of Arch Linux [2].

The contact maps (CMs) used in the experiments to validate our proposed approach – the pGEP-CA – were generated by Molecular Dynamics (MD) simulations following our previous work [2], using the 3DAB model of the protein 2gb1, which is composed by 56 amino acids. Each CM is a 56x56 matrix and represents a folding state of the folding process. In this work, 100 CMs were generated for the following threshold values: 6.65, 7, 8, 9, 10, 11 and 12Å. Thus, a total of 700 CMs were generated.

In this work, CMs are represented by 2D-CAs, which in turn, are simulated using evolved (or induced) transition rules by GEP simulations. In order to evaluate the proposed approach, experiments were done using consecutive ith and jth CMs, which in turn, represent the initial 2D-CA configuration and the expected final 2D-CA configuration, respectively. The fitness of the GEP individuals is computed using the expected CM (obtained by MD simulations) and the achieved 2D-CA built using the obtained rules. Due to the stochastic nature of the algorithm, 30 independent runs were done with different initial random seeds. Thus, a total of 20790 experiments were done.

The running parameters for the pGEP-CA are: population size ($Pop = 100$), number of GEP generations ($Maxgen = 350$), linking function ($link$ =AND), function set = [AND, OR, NOT], terminal set = [a, b, c, d, e], number of genes per chromosome ($n_g = 2$), size of head ($h = 10$), selection method sel, genetic operators probabilities ($pcross = 0.8$ – recombination, $pmut = 0.1$ – mutation, $ptIS = 0.7$ – IS transposition, $ptRIS = 0.7$ – RIS transposition, $pgt = 0.7$ – genic transposition), number of 2D-CA interactions ($CAiter = 1$), 2D-CA von Neumann neighboorhood (with $r = 1$, $m = 5$) and number of slaves ($s = 50$).

The results of our experiments are shown in Table 1. Some results are not shown here due to space restrictions. In this table, the first column shows the metrics. Next columns show their values for each threshold value. We can observe, from the $fitness$, S_C and S_{N_C} values, that the induced transition are able to generate 2D-CAs with correct contacts and non-contacts, despite the F_C and F_{NC}. It can also be observed that parallel processing was essential for obtaining results in reasonable processing time (t_p).

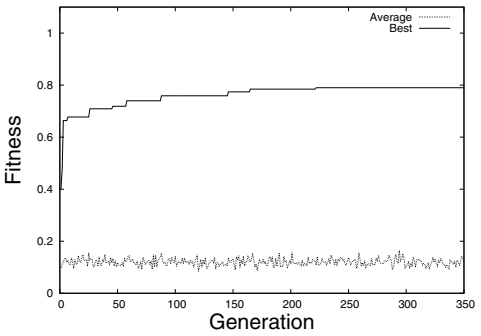
Figure 2(a) shows a plot of the fitness (best and average) obtained in a simulation. In this figure, that the genetic diversity is preserved, since the distance from average to best is maintained along GEP generations. Results can be improved hybridizing the pGEP-CA with specialized strategies in order to keep high genetic diversity and explore the search space efficiently leading to better individuals, such as local search methods or coevolution with other Evolutionary Computation algorithms as proposed by [3]. Figures 2(b) and (c) show a expected (final) CM and the obtained

[2] Available in: `https://www.archlinux.org`

CM generated using the induced rule ("$01111111011111110111111101111111_2$", with $fitness = 0.85$). The obtained CM suggest that the proposed fitness function is adequate to induce transition rules for evolving 2D-CAs, which in turn, represent CMs. We believe that best results can be obtained, using a knowledge-based strategy to correct the faults of the obtained CMs.

Table 1 Numerical results obtained using CMs with different threshold values

Metric	CM threshold [Å]			
$Avg(Min/Max)$	6.65	9	10	12
Best $fitness$	0.91 (0.73/0.97)	0.92 (0.76/0.97)	0.897 (0.76/0.93)	0.89 (0.74/0.95)
T_C	393.3 (216/428)	1323.4 (556/1392)	1548.6 (650/1644)	2160.4 (770/2280)
F_C	34.1 (12/277)	63.1 (12/211)	78.62 (20/202)	61.12 (2/182)
T_{NC}	2677.2 (2493/2789)	1682.6 (1608/2408)	1424.4 (1328/2307)	834.9 (730/2130)
F_{NC}	31.4 (12/277)	66.9 (28/206)	84.4 (32/218)	79.64 (26/338)
S_C	0.92 (0.76/0.98)	0.95 (0.76/0.98)	0.95 (0.78/0.98)	0.96 (0.74/0.988)
S_{NC}	0.98 (0.9/0.996)	0.96 (0.9/0.99)	0.95 (0.90/0.99)	0.93 (0.87/0.999)
S_i	0.9998	0.9999	0.99997	0.99999
Avg t_p(s)	12.68	12.69	12.70	12.69



(a) (b)

Fig. 2 (a) Example of performance of the pGEP-CA and Example of an obtained CM: (b) Expected CM (c) Obtained CM, where cells in states '0' and '1' are represented by white and black squares (or dots), respectively.

7 Conclusions

The process of 2D-CA transition rules induction for simulating dynamic behaviors is still an open research problem. Therefore, the approach presented in this work represents an important contribution regarding this issue.

As commented in last section, the hybridization with specialized strategies, such as local search methods and coevolution with other EC algorithms will be focused in future works.

Regarding the processing time for the simulations, future research will need highly parallel approaches for dealing with the problem, such as the use of GPGPU (General Purpose Graphics Processing Units) [15].

This work also contributes significantly to Bioinformatics, presenting the first implementation of a parallel computational approach based on GEP and CA applied to the Contact Map Prediction.

In a broader sense, we believe that the proposed approach presented in this paper is very promising for the research areas related to Cellular Automata and Protein Folding Problem.

References

1. Abu-Doleh, A., Al-Jarrah, O., Alkhateeb, A.: Protein contact map prediction using multi-stage hybrid intelligence inference systems. Journal of Biomedical Informatics 45, 173–183 (2012)
2. Benítez, C.M.V., Lopes, H.S.: Ab-initio protein folding using molecular dynamics and a simplified off-lattice model. Journal of Bionanoscience 7, 391–402 (2013)
3. Benítez, C.M.V., Parpinelli, R., Lopes, H.: A heterogeneous parallel ecologically-inspired approach applied to the 3D-AB off-lattice protein structure prediction problem. In: BRICS Countries Congress (BRICS-CCI) and Brazilian Congress (CBIC) on Computational Intelligence (2013)
4. Fariselli, P., Olmea, O., Valencia, A., Casadio, R.: Progress in predicting inter-residue contacts of proteins with neural networks and correlated mutations. Proteins 45, 157–162 (2001)
5. Ferreira, C.: Gene expression programming: a new adaptive algorithm for solving problems. Complex Systems 13, 87–129 (2001)
6. Freitas, A.: Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer (2002)
7. Lena, P.D., Nagata, K., Baldi, P.: Deep architectures for protein contact map prediction. Bioinformatics 28(19), 2449–2457 (2012)
8. Luger, G.: Artificial Intelligence: Structures and Strategies for Complex Problem Solving, 6th edn. Addison-Wesley Publishing Company, USA (2008)
9. MacCallum, R.: Striped sheets and protein contact prediction. Bioinformatics 20(1), I224–I231 (2004)
10. Matsumoto, M., Nishimura, T.: Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Transactions on Modeling and Computer Simulation 8(1), 3–30 (1998)
11. Mitchell, M.: Computation in cellular automata: a select review. In: Nonstandard Computation, 1st edn., pp. 95–140. VHC Verlagsgesellschaft, Weinheim (1998)
12. Neumann, J.V.: Theory of self-reproducing automata. University of Illinois Press (1966)
13. Peterson, W., Weldon, E.: Error-correcting codes. MIT Press (1972)

14. Topa, P., Mlocek, P.: Using shared memory as a cache in cellular automata water flow simulations on gpus. Computer Science 14, 385–401 (2013)
15. Scalabrin, M., Parpinelli, R., Benítez, C.M.V., Lopes, H.: Population-based harmony search using GPU applied to protein structure prediction. International Journal of Computational Science and Engineering 9, 106–118 (2014)
16. Weinert, W., Lopes, H.S.: Evaluation of dynamic behavior forecasting parameters in the process of transition rule induction of unidimensional cellular automata. BioSystems 99, 6–16 (2010)
17. Wolfram, S.: A New kind of science. Wolfram Media, Champaign (2002)
18. Weinert, W.R., Lopes, H.S.: GEPCLASS: A classification rule discovery tool using gene expression programming. In: Li, X., Zaïane, O.R., Li, Z.-h. (eds.) ADMA 2006. LNCS (LNAI), vol. 4093, pp. 871–880. Springer, Heidelberg (2006)
19. Mo, Y., Ren, B., Yang, W., Shuai, J.: The 2-dimensional cellular automata for HIV infection. Physica A 399, 31–39 (2014)

A Methodology to Develop Service Oriented Evolutionary Algorithms*

P. García-Sánchez, A.M. Mora, P.A. Castillo, J. González, and J.J. Merelo

Abstract. This paper proposes a methodology to design and implement Evolutionary Algorithms using the Service Oriented Architecture paradigm. This paradigm allows to deal with some of the shortcomings in the Evolutionary Algorithms area, facilitating the development, integration, standardization of services that conform a evolutionary algorithm, and, besides, the dynamic alteration of those elements in runtime. A four-step methodology to design services for Evolutionary Algorithms is presented: identification, specification, implementation and deployment. Also, as an example of application of this methodology, an adaptive algorithm is developed.

1 Introduction

New trends in Evolutionary Algorithms (EAs) [1], such such as P2P or pool-based EAs [2], lead to its implementation in dynamic and heterogeneous environments. Service Oriented Architecture (SOA) has been proposed [3] as a possible solution to facilitate the creation of applications in these kind of environments. In this paradigm, a service is a loose, coarse-grained, and autonomous component that allows interactions of all the elements that conform the system [4]. SOA could facilitate the creation and control of services for EAs in several issues addressed in other works, such as their use in *development*, as there exist several methodologies to model and design services, or the usage of techniques such as versioning, packaging or life-cycle control. Services also facilitate the *integration*, as they are independent of the programming language. Furthermore, services allow distribution transparency: it is

P. García-Sánchez · A.M. Mora · P.A. Castillo · J. González · J.J. Merelo
Dept. of Computer Architecture and Technology and CITIC-UGR
University of Granada, Spain
e-mail: pablogarcia@ugr.es

* This work has been supported in part by FPU research grant AP2009-2942 and projects SIPESCA (G-GI3000/IDIF, under Programa Operativo FEDER de Andalucía 2007-2013), PYR-2014-17 of CEI BIOTIC Granada (GENIL) and ANYSELF (TIN2011-28627-C04-02).

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_13

not mandatory to use a specific library for the distribution, or modify the code to adapt the existing operators. As SOA is based in public standards (such as WSDL[1] or OSGi[2]), its use promotes the *standardization* of the service interfaces, facilitating the Open Science [5]. Finally, as services are not aware of the order of execution, the *dynamism* of this paradigm can fit with new parallel approaches for EAs, where the control of the nodes is not centralized. For example, new operators in different nodes can be bound and used during the run of an algorithm. Also, there should be easy to add and remove elements to achieve self-adaptive mechanisms. All previous issues are taken into consideration in this paper to propose the methodology.

SOA has been previously used in the EA area. For example, web services have been used in the grid area for optimization problems [6], where services are defined using WSDL. In our previous work [3], an introduction to the usage of SOA to develop EAs, with some advantages, guidelines and examples was presented. However, previous works do not count with a guided step by step methodology for EAs, as proposed in this paper.

The rest of the paper is structured as follows: the description of the proposed methodology (called SOA-EA) is presented in Section 2. Then, in Section 3 an example of application of the methodology to create a service oriented evolutionary algorithm is performed. Finally, conclusions and future works are discussed.

2 Methodology

This section presents all the steps to design and implement Service Oriented Evolutionary Algorithms (SOEAs), that is, evolutionary algorithms whose elements are services. The selected steps have been adapted from SOMA [4], a methodology to develop services focused on business environments and adapted to be used in the EA research area.

This methodology also takes into account the work of Gagné and Parizeau [7]. The authors established six criteria to qualify the genericity of a framework for EAs: generic representation, generic fitness, generic operations, generic evolutionary model, parameter management and configurable output. Also, according to Valipour [8], services developed must follow several characteristics: they must be discoverable and dynamically bound, self-contained and modular, interoperable, loose-coupled, transparent to location and composable.

[1] `http://www.w3.org/TR/wsdl`

[2] `http://www.osgi.org/`

2.1 Identification

This phase is focused on the identification of the three constructs of SOA: services, components and flows. The developers should ask themselves several questions to facilitate the identification about the problem to solve, such as the elements and operators needed by the EA, the extension capabilities and how to parametrize of the EA. To facilitate this task three different domains are proposed. In the first one, the **algorithm domain**, the services are those that conform the EA. For example, operators of individuals, stop criterion, or populations. The second one, the **problem domain**, comprises services to address the elements of the problem (for example, the fitness function). There are also other services that depend on the problem, such as an initializer of individuals. Finally, the **infrastructure domain**, identify services that deal with the specific infrastructure that will be used to execute the algorithm. For example, services for user control, load balancing or logging. Depending on the environment where the EA is going to be developed, other services need to be modelled. For example, user control in cloud environments, different mechanisms for logging or interconnection with other systems (such as external databases).

2.2 Specification

The questions to solve prior to this phase are related with the operations and their inputs/outputs, the flow (order) of services, different kinds of available implementations and adaptation of services (metrics or behaviours).

First, the EA **operators** should not be be modelled to receive one or two individuals, but a list of individuals to be modified, as not all EAs have the same behaviour. Since many types of individuals may exist, the operators should be as abstract as possible to work properly. Therefore, services must accept interfaces of individuals as inputs, not concrete implementations, such as vectors or lists (generic representation). This is also applied to the **fitness**: it should not be calculated within a method of an *Individual* class. To be less coupled, it should be implemented as an external service that receives a list of individuals (facilitating the load balancing).

The **population** should be a service to access the individuals and allow the variation of its structure (for example, a change from an unique list population to a cellular model) without affecting the rest of the pieces of the algorithm. So, other services external to the EA could consult the *population* state and act accordingly to some rules.

Also the **parameter** set should be a service for the same reason, allowing the possibility of performing experiments related to parameter control or tuning in an efficient way (being separated from the code of the existing operators).

A SOEA can be seen as a service flow. The **flows** should be designed to reduce the impact of potential future changes. An example of service flow would be an implementation called *Evolutionary Algorithm* with all the steps common to all EAs and with independence of the implementations of these steps (generic evolutionary

model). This allow the adaptation of the evolutionary model. The user can manually select the services to be combined to create a Genetic Algorithm or an Evolution Strategy, for example. Finally, the **infrastructure services** should be designed as flexible output mechanisms. For example, a GUI or logging should be independent of the services.

2.3 Implementation and Deployment

The last two steps SOA-EA are explained together because the decisions about the technological solution to be used is bound to both phases. The questions to solve in these steps are related with the technological implementation of the services and its execution (locally or remotely). Different technologies should be compared to address the publication of interfaces, overload and dynamic control. Also, considerations about security, persistence, benchmarking and monitoring are taken into account in this step.

The first step is to **select the technology to expose the interface**, depending on the use of the services. For example, a service that is going to be used remotely and publicly from any programming language should export its interface with WSDL publicly available with an URL, to allow users to automatically generate the client for that service. On the other side, interfaces could be previously known, and it is not necessary to export them to the public. This is the case of OSGi, where the interface is exposed only to the OSGi service registry.

The **selection of the communication mechanism** must be considered depending on the system to deploy the services. In the case of EAs, where the performance is important, usually the most efficient transmission mechanism should be preferred. In this step, issues related with testing, user control, security and persistence should be taken into account.

3 Example of Creating a Service Oriented Evolutionary Algorithm

This section justifies the use of the proposed methodology and the steps to create services with it. In this example, a basic genetic algorithm to solve the MMDP problem [9] will be designed. This algorithm needs to automatically bind new operators (not previously known) when the algorithm has found a local optimum. No extra code should be added to the algorithm to bind/unbind operators or check the population.

In the *identification* step a number of abstract services have been identified. In the *algorithm domain*, the Algorithm, Population, Parent Selector, Recombinator, Mutator, Replacer, Stop Criterion and Parameters. In the *problem domain*, the Fitness Calculator and Initializer. Finally, a *infrastructure domain* service called

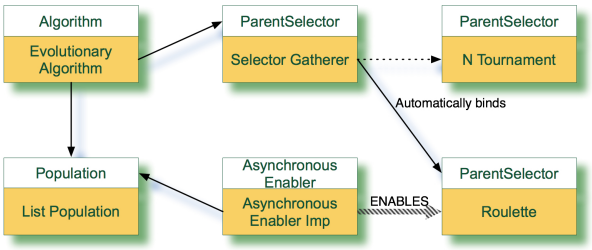
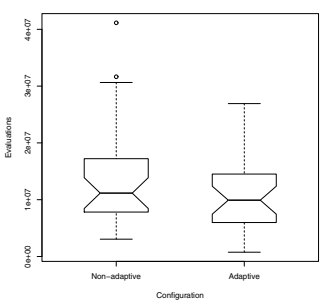


Fig. 1 Automatic binding of new operators. White boxes are interfaces and shaded boxes are implementations of the services.



	Non-adaptive
Generations	219403,10 ± 141692,16
Evaluations	14041926,40 ± 9068298,82
Time	68766,40 ± 45073,04
	Adaptive
Generations	167166,66 ± 93594,37
Evaluations	10698794,66 ± 5990039,68
Time	51710,40 ± 29329,21

Fig. 2 Results obtained. The version with automatic binding of services achieved significant better results.

Asynchronous Enabler, which is in charge of activating new operators when a local optimum is found (for example, no changes in best individual during certain time).

Concrete implementations are defined in the *specification* step: *N Tournament* and *Roulette* are implementations of parent selectors and *Optimum Found* the desired stop criterion. Also, to address the problem to be solved, such as *MMDP Fitness Calculator* or *Binary Initializer*. As we need a fixed set of steps, a *Evolutionary Algorithm* service is created to model the flow of services. The discovered services have been specified to accept a list of individuals. A service to gather all selectors in the environment is used. Infrastructure services to deal with control of the population and to enable other operators are also implemented (Asynchronous Enabler Imp). Figure 1 shows this designed configuration.

Since this example requires automatic binding and asynchronous and parallel control of the population services, OSGi is the technology proposed in the *implementation* step. The reasons of using these technologies are explained in [3], but summarizing, OSGi allows automatically binding of implementations to interfaces without extra code or recompiling, and dynamic discovery of services, as this example requires. The source code of the proposed implementation is available under a GNU/LGPL V3 license in our repository at `http://www.osgiliath.org`.

Two configuration have been compared: a non-adaptive version that only uses a Binary Tournament for Selection, and an adaptive one, which automatically enables a Roulette Selection when a local optimum is found. The parameters used in this comparison (accessed from the Parameters service) are a population of 64 individuals, selector rate of 0.5, TPX crossover, bit flip mutation, and individual length of 60 genes. The Roulette selector is enabled when the best individual of the population has not changed in 10 seconds (checked every 2 seconds). Figure 2 shows the results obtained from the 30 executions of the two configurations tested. As it can be seen, automatic and adaptive enabling of selection operators has allowed an increase of performance, reducing time and evaluations (both significantly with a p-value$<$0.05 of a Wilcoxon test). This example has been used to demonstrate that applying a methodology to develop loose coupled services that can be dynamically bound, without modification of the existing services, can be used to achieve better results.

4 Conclusions

New trends in distributed Evolutionary Algorithms, such as P2P, imply to deal with heterogeneous and dynamic environments, with different programming languages and transmission technologies. This fact motivate the creation of a proper way to define service oriented evolutionary algorithms (SOEAs) to facilitate the development, integration, standardization and dynamism of the EA components in this kind of environments. In this paper the requirements in EA design, with the requirements in SOA, have been taken into account to propose a methodology to model the services that compose a service oriented EA, and several guidelines about the design of these services have been explained. This methodology, called SOA-EA, proposes four iteratively and incremental phases: identification, specification, implementation and deployment. SOA-EA has been used to create a SOEA that takes advantage of the SOA capabilities, such as loose-coupled services and automatic binding of new operators.

In future work, this methodology will be used to create new examples of SOEAs and refined to deal with other shortcomings. Other technologies available in SOA will be also tested and analysed.

References

1. Eiben, A., Smith, J.: What is an Evolutionary Algorithm? In: Introduction to Evolutionary Computing. Springer (2003)
2. Meri, K., Arenas, M.G., Mora, A.M., Merelo, J., Castillo, P.A., García-Sánchez, P., Laredo, J.L.J.: Cloud-based evolutionary algorithms: An algorithmic study. Natural Computing, 1–13 (2013)

3. García-Sánchez, P., González, J., Castillo, P.A., Arenas, M.G., Merelo-Guervós, J.J.: Service oriented evolutionary algorithms. Soft Comput. 17(6), 1059–1075 (2013)
4. Arsanjani, A., Ghosh, S., Allam, A., Abdollah, T., Ganapathy, S., Holley, K.: SOMA: A method for developing service-oriented solutions. IBM Systems Journal 47(3), 377–396 (2008)
5. Foster, I.: Service-oriented science. Science 308(5723), 814 (2005)
6. Imade, H., Morishita, R., Ono, I., Ono, N., Okamoto, M.: A grid-oriented genetic algorithm framework for bioinformatics. New Generation Computing 22(2), 177–186 (2004)
7. Gagné, C., Parizeau, M.: Genericity in evolutionary computation software tools: Principles and case-study. International Journal on Artificial Intelligence Tools 15(2), 173 (2006)
8. Valipour, M., Amirzafari, B., Maleki, K., Daneshpour, N.: A brief survey of software architecture concepts and service oriented architecture. In: 2nd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2009, pp. 34–38 (2009)
9. Goldberg, D.E., Deb, K., Horn, J.: Massive multimodality, deception, and genetic algorithms. In: Männer, R., Manderick, B. (eds.) Parallel Problem Solving from Nature, vol. 2, pp. 37–48. Elsevier Science Publishers, B. V., Amsterdam (1992)

Simulation of Bio-inspired Security Mechanisms against Network Infrastructure Attacks

Igor Kotenko and Andrey Shorov

Abstract. The paper considers development of the approach to simulation of protection mechanisms against infrastructure attacks based on biological metaphor. The specification of models of attacks and protection mechanisms is given. The environment for security mechanisms simulation based on biological metaphor is studied, experiments that demonstrate possibilities of the designed simulation system are carried out.

1 Introduction

At present the direction of research in the area of computer networks protection against network infrastructure attacks is rather actual. Traditional protection methods often fail to cope with permanently upgrading attacks against computer systems [8]. This is the reason why researchers more often look at live nature which has efficient mechanisms of protection against malicious impact. During long evolution all living organisms developed effective mechanisms allowing them to survive and to adapt to varying environmental conditions. These mechanisms can be detected at different levels from functioning of single cell to distributed cooperative mechanisms used by nervous and immune system, whole organisms and their communities. In order to project and implement new protection systems based on biological metaphor, we need to have tools for their study, adaptation, development and testing.

Igor Kotenko
Laboratory of Computer Security Problems St. Petersburg Institute for Informatics and Automation, 14th line, 39, Saint-Petersburg, Russia
St. Petersburg National Research University of Information Technologies, Mechanics and Optics, 49, Kronverkskiy prospekt, Saint-Petersburg, Russia
e-mail: ivkote@comsec.spb.ru

Andrey Shorov
Department of Computer Science and Engineering Saint-Petersburg Electrotechnical University "LETI", Professora Popova str. 5, Saint-Petersburg, Russia
e-mail: ashxz@mail.ru

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_14

Taking into consideration the particularities of implementation of network infrastructure attacks and methods of protection against them, we suggest application of simulation methods. Simulation gives flexible mechanism of modeling of complex dynamic systems, that allows to operate with different parameters and scenarios, expending less effort than in real networks.

The paper presents the environment for development of protection mechanisms against infrastructure attacks based on bio-inspired approaches. In contrast with the paper, presented earlier [9], main attention in this paper is paid to the designed simulation environment, including models and algorithms of infrastructure attacks and mechanisms of protection against them based on biological metaphor. The main contribution of the work performed is the specification of formal model of the protection systems based on biological metaphor which is called "network nervous system" protection mechanism. Current architecture and implementation of simulation system is presented. Experiments performed on simulation of infrastructure attacks and protection mechanisms are discussed, including the "network nervous system" protection mechanism. The rest of this paper is organized as follows. Section 2 considers related work. Section 3 outlines main models and algorithms of protection mechanisms. Section 4 considers the architecture and implementation of the system for simulation of "network nervous system" protection mechanism, as well as the experiments fulfilled. Conclusion presents main results and future work directions.

2 Related Work

Research in the area of creation and implementation of approaches based on biological metaphor is rather extensive. Let us study some papers that consider bio-inspired approaches, that may be used in computer networks. M. Meisel et al. [11] suggest the classification of bio-inspired approaches by the computer networks research areas where they may be applied. "Ant" optimization algorithms is the approach used for network traffic routing optimization. On the basis of this approach the Ant-based ÑÀontrol algorythm (ABC) [14] was created. In [4] epidemic algorithms are used for actualization of distributed database.

D. Dasgupta [3] proposes a multilevel system of protection against attacks based on immunocomputing. In [5] an analogy with live cell is used to construct the mechanism of computer network protection. S. Forest and S. Hofmeyer [6] propose the approach based on the concept of an immunocomputing and negative selection. In this paper the approach called "network nervous system" and proposed by U. Chen and H. Chen [1] is studied in details. The protection mechanism based on this approach uses distributed information collecting and processing techniques which coordinate main network devices, identify attacks and implement counter-measures.

For modeling different mechanisms of infrastructure attacks and protection mechanisms researchers developed a great number of tools. These tools mainly rely on the methods for discrete event simulation of processes in network structures [12] and on trace driven models that use traces of events registered in real life computer

networks [15]. In [17], a specially developed simulation system is used to experiment with worm propagation. In [13], the simulation tool GTNetS was used to create a model of a computer worm. In [16], a simulator is developed and a botnet propagation is simulated on the model consisting of 250 thousands of nodes. Mechanisms of defense against botnet propagation are also simulated. In [7], the distributed technique for detecting DDoS attacks called Distack is simulated using the simulator OMNeT++. In [10], a specially developed simulation environment and test benches were used to evaluate the SAVE defense technique against DDoS attacks.

3 Protection Models Based on "Network Nervous System"

As component that is responsible for cooperation of base protection mechanisms the mechanism "network nervous system" is used. In every autonomous system there is special server that performs functions of information collecting and processing, coordination of the nodes connected to it and data interchange about attacks with servers in other networks. Nodes perform functions of attacks detection and blocking, as well as information transfer about detected attacks to the server to which they are connected. The architecture of the "network nervous system" and algorithms of its functioning are described in details in [9]. Here we present main components of "nervous system" at the upper level.

"Network nervous system" is represented as follows: $NN = \langle Sc, En_{NN}, Ev_{NN}, T\rangle$, where Sc is a planner of simulation system, $En_{NN} = \langle NS, NH\rangle$, where NS are servers of "network nervous system"; NH are nodes of network nervous system, e.g. routers, Ev_{NN} are events by which protection mechanisms may operate, T is modeling time.

Server of "nervous system" $NS = \langle IM, EM, DM, sDB\rangle$, where IM is the block for data interchange with nodes of "nervous system"; EM is the block for data inter-change between servers of "network nervous system"; DM is the block for decision making and determination of response reaction; sDM is database.

Server of "network nervous system" has modules for data interchange with nodes that are subordinated to it, as well as with servers that are placed in other networks. Modules of data interchange are connected to the component responsible for data analysis and decision making. Using it they receive commands and data for sending to nodes and other servers of "nervous system" and feed it with information about events that occur in the network.

The block for decision making and determination of response reaction $DM = \langle PM, CM, dEE, dBE, dAD\rangle$, where PM is module of prioritization of received data; CM is module for correlation of received data between themselves; dEE is module for data interchange with other servers of the network nervous system and nodes of local network nervous system; dBE is module for information interchange with server database; dAD is module of decision making.

Block for decision making can operate with the following events: $Ev_{DM} = \langle pM, cM, bkTr, nNTree, nSTree\rangle$, where pM is function describing work of

prioritization module; cM is function of correlation module; $bkTr$ is function of blocking module, $nNTree$ is function for processing local tree of attack source; $nSTree$ is function for processing global tree of attack source.

The node of "nervous network" $NH = \langle AG, TR, NT, HD \rangle$, where AG is module for information collection from sensors; TR is module for data interchange with server of "network nervous system"; NT is module for data interchange between nodes of "network nervous system"; HD is module implementing traffic processing.

4 Architecture, Implementation and Expirements

For implementation of the suggested models and algorithms the environment was developed that has wide range of possibilities for implementation of approaches based on biological metaphor, including "network nervous system". The system developed is a software complex that embraces several levels. The lower one is a discrete event simulation system that is implemented in low level language, and a number of components that implement entities of higher level. Models are implemented as parameterizable modules that are installed on nodes of the network simulated, on different levels (network, transport, application level). Special interface for interaction between typical nodes and modules of attack and protection is implemented for that.

The suggested architecture was implemented for simulation of infrastructure attacks (network worms propagation, DDoS) and mechanisms for protection from them. Models implemented on basis of base components of "network nervous system" are represented by server of "network nervous system" and node of "network nervous system". The simulation environment is designed on the basis of OMNeT++. For demonstration of possibilities of the simulation system the experiments on network worms propagation, DDoS attacks and protection against them were carried out. We consider only some of the experiments on protection against on network worms. To perform the experiment we built a network consisting of 3652 nodes. 1119 nodes (approx. 30% of total amount) were randomly selected as vulnerable computers. As mechanisms of counteraction to network worms propagation on all routers in the network there were subsequently installed protection mechanisms Failed Connection (FC) [2], Virus Throttling (VT) [18] and "network nervous system" (NNS). Be-sides that experiments on propagation of network worms without protection mechanisms were performed. Results of experiments showed that in case of NNS functioning the number of infected nodes decrease almost by 20% in relation to FC and approximately by 10% in relation to VT.

Fig. 1 shows the number of filtered legitimate traffic by protection mechanisms VT, FC and NNS, installed on all routers at performing attack of network worms propagation. We can see that in the beginning of the attack the number of errors of the 1st type using NNS greater than corresponding index of VT, but later this index significantly decreases. Fig. 2 shows the amount of traffic that arrive to the

attacked node, number of correctly filtered packets and number of errors of the 1st and 2nd types in case of functioning of NNS at performing attack of network worms propagation. Despite some increase of 1st type errors in the beginning of the attack the level of errors of the 2nd type remains at rather low level.

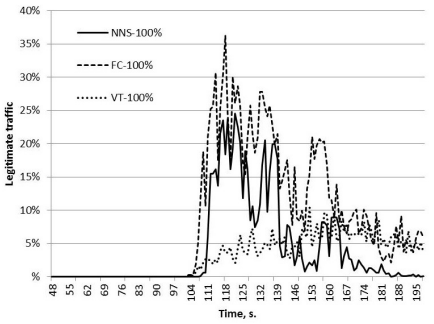


Fig. 1 Filtered legitimate traffic

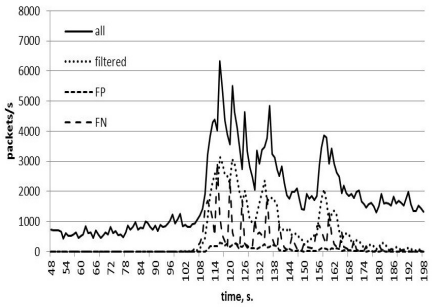


Fig. 2 Parameters of NNS

After that we compared the quality of work of different protection mechanisms. On the basis of data, received as results of experiments, there were calculated parameters of error of the 1st and 2nd type, recall (TP/(TP+FN)), precision (TP/(TP+FP)) and accuracy ((TP+TN)/(TP+FP+FN+TN)) and percentage of infected computers, where FP, FN, TP, TN are false positive, false negative, true positive and true negative rates. For FC they were respectively 0.31, 0.18, 0.52, 0.78, 0.21, 99%; for VT: 0.01, 0.57, 0.43,0.98, 0.66, 93%; and for NSS: 0.22, 0.22, 0.77, 0.90, 0.71, 81%. This shows that "network nervous system" demonstrates better performance results in comparison with other mechanisms. As we can see from the metrics presented above, the designed simulation system gives significant amount of information that can be used for analysis of efficiency of different protection mechanisms, including those based on biological metaphor.

5 Conclusions

In the paper we suggested the system for simulation of bio-inspired approaches for computer networks protection from infrastructure attacks. Formal models and algorithms of mechanisms of infrastructure attacks and mechanisms for protection from them were presented. The architecture and software prototype of simulation system was developed. Experiments on simulation of "network nervous system" protection mechanism were carried out. These experiments demonstrated possibility of efficiency analysis of work for both traditional protection mechanisms and the mechanism based on biological metaphor. Further research will be dedicated to enhancement of accuracy of security processes simulation, development of

contemporary techniques for simulation process visualization and representation of input and output data.

Acknowledgements. This research is being supported by grants of the Russian Foundation of Basic Research (projects 13-01-00843, 13-07-13159, 14-07-00697 and 14-07-00417), the Program of fundamental research of the Department for Nanotechnologies and Informational Technologies of the Russian Academy of Sciences (2.2), the state project "Organization of scientific research" of the main part of the state plan of the Board of Education of Russia as well as by Government of the Russian Federation, Grant 074-U01, and State contract #14.604.21.0033.

References

1. Chen, Y., Chen, H.: NeuroNet: An Adaptive Infrastructure for Network Security. International Journal of Information, Intelligence and Knowledge 1(2) (2009)
2. Chen, S., Tang, Y.: Slowing Down Internet Worms. In: Proc. of the 24th International Conference on Distributed Computing Systems (2004)
3. Dasgupta, D.: Immuno-inspired autonomic system for cyber defense. Information Security Tech. Report archive 12(4) (2007)
4. Demers, A., Greene, D., Hauser, C., et al.: Epidemic algorithms for replicated database maintenance. In: Proc. of the Sixth Annual ACM Symposium on Principles of Distributed Computing, PODC 1987 (1987)
5. Dressler, F.: Bio-inspired mechanisms for efficient and adaptive network security. Service Management and Self-Organization in IP-based Networks (2005)
6. Hofmeyr, S., Forrest, S.: Architecture for an artificial immune system. Evolutionary Computation 8(4) (2000)
7. Gamer, T., Mayer, C.: Large-Scale Evaluation of Distributed Attack Detection. In: Proc. of the 2nd Int. Workshop on OMNeT++, Rome (2009)
8. Kotenko, I., Konovalov, A., Shorov, A.: Agent-based Modeling and Simulation of Botnets and Botnet Defense. In: Conference on Cyber Conflict. CCD COE Publications (2010)
9. Kotenko, I., Shorov, A., Novikova, E.: Simulation of Protection Mechanisms Based on “Nervous Network System" against Infrastructure Attacks. In: Proc. of the 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP 2013). IEEE Society (2013)
10. Li, J., Mirkovic, J., Wang, M., Reither, P., Zhang, L.: Save: Source address validity enforcement protocol. In: Proc. of IEEE INFOCOM (2002)
11. Meisel, M., Pappas, V., Zhang, L.: A Taxonomy of Biologically Inspired Research in Computer Networking. Computer Networks (2009)
12. Owezarski, P., Larrieu, N.: A Trace Based Method for Realistic Simulation. In: Comm. of the IEEE Int. Conf., Toulouse (2004)
13. Riley, G., Sharif, M., Lee, W.: Simulating Internet Worms. In: Proc. of the 12th Int. Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Atlanta (2004)
14. Schoonderwoerd, R., Bruten, J., Holland, O., Rothkrantz, L.: Ant-based load balancing in telecommunications networks. Adapt. Behav. 5(2) (1996)
15. Simmonds, R., Bradford, R., Unger, B.: Applying Parallel Discrete Event Simulation to Network Emulation. In: Proc. of the Fourteenth Workshop on Parallel and Distributed Simulation (PADS 2000), Washington (2000)

16. Schuchard, M., Mohaisen, A., Kune, D., et al.: Loosing Control of the Internet: Using the Data Plane to Attack the Control Plane. In: Proc. of the 17th ACM Conf. on Computer and Communication Security (CCS 2010). ACM (2010)
17. Wagner, A., Dubendorfer, T., Plattner, B., et al.: Experiences with Worm Propagation Simulations. In: Proc. of the ACM Workshop on Rapid Malcode, New York (2003)
18. Williamson, M.: Throttling Viruses: Restricting propagation to defeat malicious mobile code. In: Proceedings of ACSAC Security Conference (2002)

Part V
Cloud and Grid Computing

Improving Grid Nodes Coalitions by Using Reputation

Pasquale De Meo, Fabrizio Messina, Domenico Rosaci, and Giuseppe M. L. Sarné

Abstract. In this work we deal with the issue of improving the QoS provided by each node of a Grid Federation, by modelling it as a problem of "Grid formation". In the proposed model each Grid node belonging to a computational Grid, is free to join with or leave a grid with the goal of improving its satisfaction. Contextually, each grid is free to search other nodes to join with it or to remove those nodes resulted ineffective. Software agents manage the node profiles and in our model a *Grid agent* has the role of handling the profile of the Grid. We introduce a distributed algorithm, called GF, to handle the node activity of joining to the grid, modelled as a matching problem. Some experiments shown the effectiveness of our approach.

Keywords: Grid Computing, Multi-agent systems, QoS.

1 Introduction

The Grid Computing paradigm, widely adopted in the last years, extended the conventional distributed computing paradigm to large-scale resource sharing [5]. The recent development of Federated Grids [7] allows institutions to share resources among different types of grid infrastructures. Besides, Federated Grids are enhanced and supported by the use of virtualization technology [9], by which Grid nodes are configurable in a flexible manner. Therefore Grid Federations assumes a highly

Fabrizio Messina
DMI, University of Catania, V.le A. Doria 6, Catania, Italy
e-mail: messina@dmi.unict.it

Pasquale De Meo
DICAM, University of Messina, 98166 Messina, Italy
e-mail: pdemeo@unime.it

Domenico Rosaci · Giuseppe M.L. Sarné
{DIIES, DICEAM}, University "Mediterranea" of Reggio Calabria,
Loc. Feo di Vito, 89122 Reggio Calabria, Italy
e-mail: domenico.rosaci,sarne@unirc.it

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_15

dynamic nature as companies and institutions can easily enter or leave grid infrastructures, changing their own role, adding, removing or reconfiguring their nodes very quickly.

Basing on the considerations above, in this work we deal with the problem of maximizing the *Quality Of Service* (QoS) associated with each node in providing a service in a Grid Federation, by modelling it as a "Grid formation" problem, where each node, in order to improve its satisfaction, is free to join with or leave a grid. Contextually, each grid is free to search other nodes to join with or to remove those resulted ineffective. We further assume that the profile of each Grid node is managed by a software agent [6] and that the set of nodes belonging to each Grid is coordinated from a *Grid agent* managing the Grid profile. Moreover we model the process of joining with a grid for a node and the process to accept a node by a computational Grid by means of a matching between the behaviours of nodes and the behaviours of the grids in a distributed fashion. To this purpose, we define some performance measures and an algorithm, named *Grid Formation* (GF), designed to find a suitable matching between nodes and grids by improving both individual and global satisfaction measures into the grids. The experimental evaluation of the GF algorithm, carried out on a set of simulated nodes and grids, clearly shows the advantages of our proposal.

The plan of the paper is as follows. Section 2 introduces the reference scenario and Section 3 describes the main behaviours/tasks of grid and node agents. Sections 4 and 5 present the GF algorithm and the performed experiments, respectively. Finally, in Section 3 we discuss the related work and in Section 7 we draw our conclusions.

2 The Reference Scenario

Let $\mathcal{N}$ be the space of k computational nodes, and $\mathcal{G}$ the space of the m grids formed with nodes in $\mathcal{N}$. As usual, each Grid $g_j \in G$ provides computational and storage services offered by any of its nodes. We suppose that each node n_i is associated with a software agent a_i supporting its activities, and each node managing a grid g_j is associated with a software agent helping it in administrating the grid, say A_j.

In such a context we model and study the evolution of the Grid Federation G by modelling the behaviour of the generic node $n_i \in \mathcal{N}$, in terms of *joining* or *leaving* the grid g_j, and the behaviour of the grid g_j, in terms of accepting or rejecting the generic node n_i. This is done based on three different measures discussed below, i.e. (*i*) the *behavioural measures*, the (*ii*) *reputation measures* and (*iii*) the *convenience measures* managed by the agents of the nodes (grid) to which they are assigned.

2.1 Behavioral Measures

The behavioural measures definitions rely on the consideration that nodes (grids) characterised of a significant need of resources will have to look for computational grids (nodes) having suitable capabilities to satisfy such needs and vice versa.

We define the *Resource cost* R as the price of a service with respect to the amount of offered/consumed computational and storage resources:

$$R = c_C \cdot C + c_S \cdot S \tag{1}$$

where C is the computational capability of a node, expressed in $MIPS$ (Million Instructions Per Second), S represents its storage capability, expressed in TB (Terabyte), c_C represents the unitary cost of the offered computational capability and c_S is the unitary cost of the storage capability.

Thereafter, the "historical" attitude of a node to offer or to consume resources within a Grid is defined as the *Node Behaviour* (NB) and measured as:

$$NB = \alpha \cdot NB + (1 - \alpha) \cdot \frac{R^{req}}{R^{req} + R^{off}} \quad (\alpha \in [0, 1] \in \mathbb{R}) \tag{2}$$

where R^{req} and R^{off} are the costs of requested and offered resources, respectively. The new value of NB ($\in [0, 1] \in \mathbb{R}$), as expressed in Eq. (2) is computed by weighting, by the parameter α, its current value and a second contribution due to the new services for which the node has been involved as provider and/or consumer. The higher the value of α the greater the contribution of the current value of NB and vice versa. The second contribution is calculated by the costs of the involved requested (R^{req}) and offered (R^{off}) resources. $NB = 1$ means the node tends to require resources to the other nodes belonging to the grid, while $NB = 0$ means the node is generally self-sufficient and tend to offer its own resources to the other nodes. Note that the updating rule used in Eq. (2) is often exploited in multi-agent systems, for instance in [24, 25], with good results.

We are also interested in the evaluation of the current *global footprint* of a grid in offering or consuming resources, therefore we define the *Grid Behaviour* (GB) as the average of the NB measures of all the nodes joined to the specific grid, say g_k:

$$GB = \frac{1}{z} \sum_{v=1}^{z} NB_v \quad (z = ||g_k||) \tag{3}$$

Clearly GB assumes a value in $[0, 1]$, and represents the tendency of the whole Grid to offer or require resources.

2.2 The Reputation Measures

The second introduced measure is based on the concepts of *feedback* and *reputation*. We assume that reputation is "*an expectation about the user's behaviour based on information about or observations of his past behaviour*" [1, 23]. Besides, our approach is inspired to [10], which aimed to promote the QoS by (*i*) considering the whole history of each node in offering services and (*ii*) detecting possible manipulations of the reputation [11].

In the following, we assume that the measure of reputation computed for a given node n_j, say ρ_j, ranging in $[0, 1] \in \mathbb{R}$, where $0/1$ means that the node is totally unreliable/reliable. Moreover the initial reputation of new nodes is set as 0.5 to penalise them for not too much but enough to contrast whitewashing strategies.

We also assume that, after a node n_i has provided another node n_j with a service $s_{i,j}$ having a resource cost $R_{s_{i,j}}$, then n_i provides a feedback, say $\phi_{s_{i,j}} \in [0, 1] \in \mathbb{R}$, about the QoS perceived about the provisioning of service $s_{i,j}$, where $0/1$ means that n_j is totally unsatisfied/satisfied for $s_{i,j}$. The reputation ρ_j for n_j is defined as:

$$\rho_j = \beta \cdot \rho_j + (1 - \beta) \cdot \phi_{s_{i,j}} \tag{4}$$

where

$$\beta = \left(\gamma \cdot \delta_{ij} + (1 - \gamma) \cdot \frac{R_{s_{i,j}}}{R_{max}}\right) \cdot \epsilon_i \qquad \delta_{ij} = \frac{\rho_i}{\rho_j + \rho_i} \cdot \left(1 - \frac{\phi_i^-}{\phi_i^- + \phi_i^+}\right) \tag{5}$$

Basing on Eq. (4), the reputation ρ_j of the generic node n_j is updated by combining two contributions, the former is the previous value of ρ_j while the latter is the feedback provided by n_i weighted by the coefficient β, which takes values in $[0, 1]$.

In computing β, in the left part of Eq. (5), several factors have been taken into account. First of all, *in order to avoid false feedbacks*, *reliability* of node n_j (i.e. δ_{ij}, specified in the right part of Eq.5) is weighted by means of a factor $\gamma \in [0, 1] \in \mathbb{R}$. Moreover, in computing δ_{ij} , ϕ_i^-/ϕ_i^+ is the number of negative/positive feedback provided by the node n_i, therefore δ_{ij} is high when $\rho_i >> \rho_j$ and the number of negative feedbacks provided by n_i (i.e. ϕ_i^-) is negligible with respect to the total number of feedbacks provided by n_i (i.e. $\phi_i^- + \phi_i^+$), which is the desired property.

Secondly we introduced in (5) the cost of resources ($R_{s_{i,j}}$) as the weighted ratio between the Resource cost of the service $s_{i,j}$ and R_{max} , which is a system parameter representing the maximum possible resource cost value in $\mathcal{N}$. This is done to not allow nodes of gaining reputation on services having a low resource cost value R.

Finally, to contrast collusive behaviours devoted to increase or decrease the reputation of a node by providing a number of (positive or negative) feedbacks, the parameter Collusion (ϵ) is adopted and computed as $\epsilon = S_{i,j}^{-e}$, where $S_{i,j}$ is the number of previous services occurred between n_i and n_j and e is a non-negative value. In particular, $\epsilon = 1$ only for $S_{i,j} = 1$, i.e. the first time that n_i and n_j reciprocally interact, the coefficient β is not decreased, while when $S_{i,j}$ raises ϵ becomes closer to 0, depending on the value of e, and decreases β.

2.3 The Convenience Measures

In order to measure the *convenience* for a node n_i (grid g_k) to join with (to accept the request of) the grid g_k (the node n_i) we define the measure $\gamma_{i,k}$ ($\eta_{k,i}$) as follows:

$$\gamma_{i,k} = |NB_i - GB_k| \cdot \frac{\sum_{n_l \in g} \rho_l}{||g_k||} \qquad \eta_{k,i} = |NB_i - GB_k| \cdot \rho_i$$

In words, the convenience $\gamma_{i,k}$ for n_i to join with g_k increases with the difference between the behaviours of n_i and g_k, and with the average reputation of all the nodes belonging to g_k. Analogously, the convenience $\eta_{k,i}$ for a grid g_k to add a given node n_i as a measure that takes into account the Behaviours of n_i and g_k, on the one hand, and the reputation of n_i on the other hand.

3 The Agents

In the proposed model nodes and grids, are supported in their activities by software agents associated with. In the following we synthetically describes the representations of knowledge of such agents as well as their behaviours.

The Agent Knowledge. We define the knowledge, or profile, p_n (resp. p_g) that an agent holds and manages about a node (resp.a grid) as a tuple having the form of $\langle WP, BP, KR, C \rangle$, where: (*i*) WP is the *Working Profile* storing the amount of resources of its associated node (grid); (*ii*) BP is the *Behaviour Profile* storing behavioural measures (see Section 2.1) of its associated node (resp. grid); (*iii*) KR is the *Key Repository* storing a pair of keys for asymmetric cryptography; (*iv*) C is the *Certificate* storing (*a*) the last reputation score of the node (resp. grid) built with the feedbacks provided by agents exploited its services; and (*b*) the value of its last NB (resp. GB) measure computed by its last grid agent (resp.the same grid agent). This certificate is signed with the public key of its last grid agent (resp. the same grid agent) in order to preserve its stored information from malicious behaviours.

The Node Agent Tasks. Each node agent (*i*) manages the node profile and resources; (*ii*) requires to join with (i.e. leaves) a grid belonging to $\mathcal{N}$ by evaluating the grid-node compatibility for reciprocal behavioural and reputation rates, stored in their certificates, as specified into Section 4; (*iii*) requires (resp. accepts/declines) a service (resp. to provide a service) to another node belonging to its grid based on a *Service Level Agreement* (SLA) (resp. a SLA proposed by a grid agent); (*iv*) sends a feedback (see Section 2.2) to its grid agent about the node provider of a required and exploited service considering its perceived QoS on the basis of an agreed SLA.

The Grid Agent Tasks. The grid agent (*i*) manages the grid profile taking into account the resources made available from its affiliated nodes; (*ii*) requires to a node of joining with/leaving the grid or accept a node which has requested to join with the grid (see Section 4); (*iii*) collect the feedbacks of each node which has

provided a service to another one its node; (*iv*) updates behavioural information (see Section 2.1) of the grid basing on the information received into the feedbacks provided by the nodes; (*v*) updates and signs (with its private cryptographic key) the certificates of the grid and of the nodes involved in a service task.

4 The GF Algorithm

In this section we define a decentralised procedure, named GF (Grid Formation), composed by a set of activities periodically executed by each node agent a_n (resp., grid agent a_g) and based on the measures defined in Section 2. To this purpose, let T be the time elapsed between two consecutive executions of the GF algorithm, that we define as *epoch*. Moreover, we assume that agents can query a distributed database named GR (Grid Repository) on which the list of the grids is maintained.

GF Tasks Performed by the Node Agent. Let X_n be the set of the grids a_n is affiliated to, where $||X_n|| \leq N_{MAX}$, and N_{MAX} is the maximum number of grids a node agent can join with. We suppose that a_n, for each grid contacted in the past, stores into a cache its grid profile p (see Section 3) and the date d of the last time that a_n executed GF for that grid. Finally, let ψ_n be a date and $\tau_n \in [0, 1]$ be thresholds fixed by the agent a_n. The GF tasks performed by the agent a_n for the grid g are defined as follows:

1. A set Y of N_{max} grids from GR, so that $X_n \bigcap Y = \{0\}$ is randomly selected.
2. For each grid $g \in Z = (X_n \bigcup Y)$ such that $d_g > \psi_n$, a message to the agent a_g to ask the profile p_g is sent.
3. For each received p_g, the convenience measure $\gamma_{n,g}$ (see Section 2.3) between its profile and that of the grid g is computed.
4. A list L_{good} with all the grid $g \in Z$, such that $\gamma_{n,g} > \tau_n$ is filled.
5. A second list L'_{good} builds by inserting a number $n = min(N_{max}, ||L_{good}||)$ grids of L_{good} with the greater value of $\gamma_{n,g}$ is filled.
6. For each grid $g \in L'_{good}$, if $g \notin X_n$, a join request to a_g together with the profile p_n is sent.
7. For each $g \in X$, $g \notin L'_{good}$, then a_n deletes n from g, i.e. a message to a_g in order to leave the grid g is sent.

GF Tasks Performed by the Grid Agent. Let K_g be the set of the nodes affiliated to the grid g, where $||K_g|| \leq K_{MAX}$, being K_{MAX} the maximum number of nodes allowed by the administrator of g. Suppose that a_g stores into its cache the profile p_u of each node $u \in K_g$, let d_u the date of its acquisition. Finally, Let ω_g a date and $\pi_g \in [0, 1] \subset \mathbb{R}$ two thresholds fixed by agent a_g. The following GF tasks are performed by the grid agent a_g and triggered whenever a join request by a node agent a_n (along with its profile p_n) is received by a_g:

1. For each node $u \in K_g$ such that $d_u > \omega_g$, a message is sent to the agent a_u to require the profile p_u associated with u.
2. The convenience measure $\eta_{g,u}$ (see Section 2.3) for each node $u \in K_g \bigcup \{n\}$ and the profile of the grid g is computed.
3. A list K_{good} is filled with the nodes u such that $\eta_{g,u} > \pi_g$, are inserted in the list K_{good} of good candidates.
4. A list K'_{good} is filled with a number $n = min(K_{max}, ||K_{good}||)$ of nodes from K_{good} having the greater value of $\eta_{g,u}$.
5. For each node $u \in K_g$, if $u \notin K'_{good}$, a_g deletes u from g. Clearly, if $n \in K'_{good}$, its request to join with g is accepted.

5 Experiments

In this section we describe some experiments we performed to evaluate the effectiveness of the GF algorithm. To this purpose, we implemented a simulator written in JAVA, capable of simulating the activities of agent nodes and grid nodes.

All the experiments we present here have been conducted on a simulated scenario having 2.000 nodes and 40 grids, where each node and each grid were provided with a user profile as described in Section 2. In particular, each profile p_n of a user n has been generated by assuming that NB_n and ρ_n are sampled from a uniform random distribution. Each node was randomly assigned to a number of grids comprised between 2 and 10. The values of the thresholds and parameters introduced in Section 2.2 and 4 are shown in Table 1. These values were selected as those which produced the best results in a sensitivity analysis we have performed.

Table 1 Values of the parameters and thresholds used in the simulator

	β	τ	π	$KMAX$	$NMAX$	$NREQ$
Value	0.7	0.4	0.4	80	10	5

As a measure of the internal *convenience* of a grid g_k, we use the concept of *Average Convenience* (AC_k), defined as the average of the convenience values $\eta_{k,i}$ computed on all the nodes $n_i \in g_k$. In order to measure the global convenience of the grids in our simulated scenario, we compute the mean (MAC) and the standard deviation (DAC) of all the AC_k, by the formulas:

$$MAC = \frac{\sum_{g_k \in G} AC_k}{|G|} \qquad DAC = \sqrt{\frac{\sum_{g_k \in G} (AC_k - MAC)^2}{|G|}}$$

In our simulation, after profiles of nodes and grids were generated, the initial values for the above measures were $MAC = 0.157$ and $DAC = 0.004$, indicating a population with a very low level of convenience. This is due to the generation of the

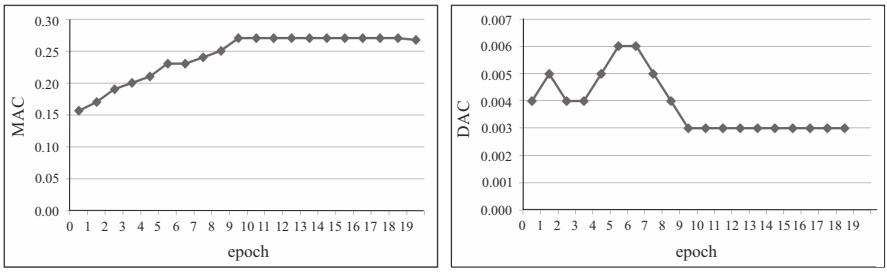


Fig. 1 Variation of MAC (left) and DAC (right) vs epochs

grids which was completely random. Then we applied the GF algorithm described in Section 4, simulating a number of 20 epochs (see Section 4) of execution for each node. The results of the simulation, in terms of MAC (DAC) with respect to the epochs, are shown in the left (right) part of Figure 1.

The results clearly show that the GF algorithm introduces a significant increment of the convenience of the grids, that after a period of 9 epochs achieves a stable configuration in which $MAC = 0.267$ and $DAC = 0.0033$. Therefore, we have obtained an improvement of about a 30% in average convenience of the grids, while the standard deviation from this compactness value remains small enough.

6 Related Work

Currently two emerging needs exist in the grid context. The first one involves the group formation in a competitive grid scenario, for instance as in [15,17], where authors consider deriving their approach from comparable contexts [2, 4, 8, 16, 17, 19, 26]. The other one is referred to connect different Grids [5] everywhere distributed into federations, it has been widely considered as a problem of inter-grid scheduling and coordination. Authors of [21] present a model of Grid Federation as an economy driven large scale scheduling system, and propose a resource allocation algorithm to improve the scheduling of cluster resources across the grid federation by satisfying the QoS constraints dictated by the resource consumers. In [20] a federation of grid agents adopting a "computational economy methodology" is discussed as a solution to couple together distributed cluster resources in a dynamic and cooperative environment in order to facilitate the overall QoS scheduling. Authors of [22] deal with the problem of conflicting schedules between different distributed workflow brokers, by proposing a Peer-to-Peer approach to coordinate the the Grid wide distributed workflow brokers, and provide a trace driven simulation study by which they prove the effectiveness of the approach in avoiding conflicting schedules. The works cited above do not deal with the problem to optimise the formation of dynamic grids in federation not considering its dynamic nature. In [27] the technology for the federation of grid is used to build an utility computing infrastructures

to provide a full metascheduling system which is flexible, scalable and based on standards. The authors claims as their solution exhibits many advantages (security, scalability, etc.) and good performances, especially with compute-intensive applications. Three grid resource allocation mechanisms based on the game theory are compared and analysed in [8]. Cooperative, semi-cooperative and non-cooperative approaches are discussed and their performances evaluated by an extended set of simulations. All the cited approaches mainly deals with the problem of scheduling and resource management in Grid Federations, while by our approach we aim at reaching an equilibrium between requested and offered resources within each grid.

7 Conclusions and Future Work

This papers deals with the problem of maximizing the QoS provided by each node in a Grid Federation. In our scenario we suppose that each node is free to join with or leave a grid as well as to start (resp. stop) collaborating with other nodes. Software agents are exploited to manage the profile of each node; in our model we introduce a *Grid agent* having the role of handling the profile of the Grid. We propose a distributed algorithm, called Grid Formation (GF), to optimize the node activity of joining with the grid. In particular, the GF algorithm solves a matching problem between the behaviours of nodes and the behaviours of the grids. We have presented some experiments showing the effectiveness of our approach in improving the convenience of the grid when accepting new nodes.

As for future work we plan to test our algorithm in presence of very large Grids (millions of nodes) by using a new software simulator for complex networks [18–20] and to provide a theoretical foundations about the scalability of our approach.

Acknowledgements. This work is a part of the research project **PRISMA**, code **PON04a2_A/F**, funded by the Italian Ministry of University within the **PON 2007-2013** framework program.

References

1. Abdul-Rahman, A., Hailes, S.: Supporting trust in virtual communities. In: HICSS, vol. 6. IEEE Computer Society, Washington, DC (2000)
2. De Meo, P., Ferrara, E., Rosaci, D., Sarnè, G.M.L.: How to improve group homogeneity in online social networks. In: Proceedings of the 14th WOA 2013. CEUR Workshop Proceedings, vol. 1099. CEUR-WS.org (2011)
3. De Meo, P., Nocera, A., Terracina, G., Ursino, D.: Recommendation of similar users, resources and social networks in a social internetworking scenario. Information Sciences 181(7), 1285–1305 (2011)
4. De Meo, P., Quattrone, G., Ursino, D.: Integration of the hl7 standard in a multiagent system to support personalized access to e-health services. IEEE Trans. on Knowledge and Data Engineering 23(8), 1244–1260 (2011)
5. Foster, I., Kesselman, C.: The Grid 2: Blueprint for a new computing infrastructure. Access Online via Elsevier (2003)

6. Jennings, N.R., Wooldridge, M.: Applying agent technology. Applied Artificial Intelligence 9(4), 357–369 (1995)
7. Katia, L., Eduardo, H., Ignacio, M.L.: A decentralized model for scheduling independent tasks in federated grids. Future Generation Computer Systems 25(8), 840–852 (2009)
8. Khan, S.U., Ahmad, I.: Non-cooperative, semi-cooperative, and cooperative games-based grid resource allocation. In: 20th International Parallel and Distributed Processing Symposium, IPDPS 2006, 10 p. IEEE (2006)
9. Kivity, A., Kamay, Y., Laor, D., Lublin, U., Liguori, A.: KVM: the linux virtual machine monitor. In: Proc. of the Linux Symp., vol. 1, pp. 225–230 (2007)
10. Lax, G., Sarné, G.M.L.: CellTrust: a reputation model for C2C commerce. Electronic Commerce Research 8(4), 193–216 (2006)
11. Massa, P.: A survey of trust use and modeling in current real systems. In: Trust in E-Services: Technologies, Practices and Challenges. Idea Group Publishing (2006)
12. Messina, F., Pappalardo, G., Rosaci, D., Santoro, C., Sarné, G.M.L.: HySoN: A distributed agent-based protocol for group formation in online social networks. In: Klusch, M., Thimm, M., Paprzycki, M. (eds.) MATES 2013. LNCS, vol. 8076, pp. 320–333. Springer, Heidelberg (2013)
13. Messina, F., Pappalardo, G., Rosaci, D., Santoro, C., Sarné, G.M.L.: A trust-based approach for a competitive cloud/Grid computing scenario. In: Fortino, G., Badica, C., Malgeri, M., Unland, R. (eds.) IDC 2012. SCI, vol. 446, pp. 129–138. Springer, Heidelberg (2012)
14. Messina, F., Pappalardo, G., Rosaci, D., Santoro, C., Sarné, G.M.L.: A distributed agent-based approach for supporting group formation in P2P e-learning. In: Baldoni, M., Baroglio, C., Boella, G., Micalizio, R. (eds.) AI*IA 2013. LNCS, vol. 8249, pp. 312–323. Springer, Heidelberg (2013)
15. Messina, F., Pappalardo, G., Santoro, C.: Complexsim: a flexible simulation platform for complex systems. International Journal of Simulation and Process Modelling, 8(4), 202–211
16. Messina, F., Pappalardo, G., Santoro, C.: Complexsim: An smp-aware complex network simulation framework. In: 2012 Sixth International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), pp. 861–866 (2012)
17. Messina, F., Pappalardo, G., Santoro, C.: Decentralised resource finding in cloud/grid computing environments: A performance evaluation. In: WETICE, pp. 143–148
18. Messina, F., Pappalardo, G., Santoro, C.: Exploiting gpus to simulate complex systems. In: 2013 17th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS), pp. 535–540 (2013)
19. Messina, F., Pappalardo, G., Santoro, C.: Hygra: A decentralized protocol for resource discovery and job allocation in large computational grids. In: 2010 IEEE Symposium on Computers and Communications (ISCC), pp. 817–823 (2010)
20. Ranjan, R., Buyya, R., Harwood, A.: A case for cooperative and incentive-based coupling of distributed clusters. In: IEEE Int. Cluster Computing, pp. 1–11 (September 2005)
21. Ranjan, R., Harwood, A., Buyya, R., et al.: Grid federation: An economy based, scalable distributed resource management system for large-scale resource coupling. In: Grid Computing and Distributed Systems Laboratory, University of Melbourne, Australia (2004)
22. Ranjan, R., Rahman, M., Buyya, R.: A decentralized and cooperative workflow scheduling algorithm. In: 8th Int. Symp. on Cluster Computing and the Grid, pp. 1–8. IEEE (2008)
23. Rosaci, D.: Trust measures for competitive agents. Knowledge-Based Systems 28, 38–46 (2012)
24. Rosaci, D., Sarné, G.M.L.: Efficient personalization of e-learning activities using a multi-device decentralized recommender system. Computational Intelligence 26(2), 121–141 (2010)
25. Rosaci, D., Sarné, G.M.L.: Recommending multimedia Web services in a multi-device environment. Information Systems 38(2), 198–212 (2013)
26. Rosaci, D., Sarné, G.M.L.: Matching users with groups in social networks. In: Zavoral, F., Jung, J.J., Badica, C. (eds.) IDC 2013. SCI, vol. 511, pp. 45–54. Springer, Heidelberg (2013)
27. Vázquez, T., Huedo, E., Montero, R.S., Llorente, I.M.: Evaluation of a utility computing model based on the federation of grid infrastructures. In: Kermarrec, A.-M., Bougé, L., Priol, T. (eds.) Euro-Par 2007. LNCS, vol. 4641, pp. 372–381. Springer, Heidelberg (2007)

User-Centric Cloud Intermediation Services

Luís Nogueira and Jorge Coelho

Abstract. Given the wide adoption of cloud computing technology and the growing number of service providers, consumers will increasingly face the challenge of finding appropriate providers that can satisfy their functional and non-functional requirements. In this paper, we examine the exploitation of machine learning techniques in user modelling technology for cloud services as a way to inform service providers on how to target service configurations at specific market segments, as well as to advice consumers on their service requests. More specifically, we examine the organisation of consumers into meaningful subsets (communities) and the extraction of meaningful service configurations for each subset (stereotypes).

1 Introduction

Despite the marketing hype touting unfettered access to cloud services as a benefit, the success of new cloud solutions that costumers are increasingly demanding will be greatly affected by their efficiency in the capability to satisfy stricter requirements than just basic service availability or best-effort computation.

In this context, the use of efficient cloud brokering mechanisms are essential to transform the heterogeneous cloud market into a commodity-like service [2, 7, 9]. By increasing the degree and the sophistication of the intermediation process, cloud commerce becomes much more dynamic, personalised, and context sensitive. From the customer's perspective, it is desirable to have an entity that could search all the available offers to find the most suitable one, and then go forward through the

Luís Nogueira
ISEP/CISTER/INESC-TEC, Portugal
e-mail: lmn@isep.ipp.pt

Jorge Coelho
ISEP/LIACC, Portugal
e-mail: jmn@isep.ipp.pt

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_16

process of actually negotiate technical contracts on-the-fly, delivering the high levels of flexibility consumers are now demanding. From the service provider's perspective it is desirable to vary its own offering depending on the customer it is dealing with, on what its competitors are doing, and on the current state of its own business.

A recent report [6] surveys the research outcomes stemming from European projects, discusses how these outcomes address the complete SLA lifecycle and provides recommendations, not only technological, but also covering legal, economic and standardisation areas. One of the main conclusions is the need to increase the cloud market pool for non-technical users through simplicity, relieving the user from the need to be aware of all service and infrastructure parameters". However, none of the reviewed projects is presented as a potential contribution to fulfil this goal. This is the main goal of the work we present in this paper.

We focus on unsupervised learning techniques for grouping potential consumers in meaningful classes we call *communities*. Then, we extract a meaning from each generated community, that is, associate its elements (consumers) with a limited set of common service interests, we call a *stereotype*. The overall approach not only allows a broker to inform service providers on how to target service configurations at specific market segments, but also to advice consumers to the typical service configuration, selected by consumers with similar characteristics and needs, and learned in previous interactions with the system.

2 Modelling Consumers to Provide Advice

As the cloud service model matures and becomes ubiquitous, it raises the possibility of improving the way services are provisioned and managed, allowing providers to address the diverse needs of consumers. In this context, Service Level Agreements (SLAs) emerge as a key aspect. Besides setting the expectations of consumers by dictating the quality and the type of service, SLAs are also increasingly considered by service providers as the key differentiator to achieve competitive advantage and the mean to establish their credibility [6].

At the same time, interest has grown in discovering user communities as a way to provide a segmentation of consumers, moving the focus from the individual to the communities in which he or she belongs, albeit mostly with approaches that cannot address the complexity and dynamism of the cloud computing environment [8].

On that account, this paper proposes a method for constructing consumer communities and their respective stereotypes in dynamic environments, which at its core applies a modified version of the COBWEB unsupervised conceptual clustering algorithm [3]. Fisher showed that, by using the concept of category utility [4], a highly effective probabilistic conceptual clustering algorithm could be produced. The category utility calculation takes account of each attribute in an instance, comparing it to the attributes of the instances within a given cluster, returning the utility as a measure of how much information they have in common. This attribute-value pair

comparison used within COBWEB is able to create a clustering that maximises inter-cluster dissimilarity and intra-cluster similarity.

In our case, we are using COBWEB as a tool for grouping consumers with similar characteristics in meaningful clusters we call *communities*. Each concept in the hierarchy produced by COBWEB is, therefore, a probabilistic structure that summarises the objects classified under that concept. Note that communities are built from data containing only the characteristics of consumers, which are determined by the consumers themselves.

On the other hand, in order to provide advice to consumers on their service requests, a representative service configuration of its community must be determined. However, this can be done effectively only when the communities are meaningful. Thus, in addition to classifying consumers into communities, there is the need to associate consumers with a limited set of common interests in the characteristics of the obtained SLAs at the end of the negotiation process.

COBWEB in its original form was not intended to have updates occur to the instances that were present within the tree structure. The tree is sorted based on the likeness of the objects resident within it and to locate a given instance each instance would have to be examined in turn, resulting in a $O(n)$ search time. Further, a modification of an instance within the tree itself would change the category utility of the node containing it, and any parent nodes, thus destroying the integrity of the tree.

Therefore, we propose a faster way of searching the COBWEB tree in order to enable updates of each consumer in the tree to occur in a more efficient manner and without changing the integrity of the tree. For that, an index for the COBWEB tree is implemented through an additional red-black tree [5]. For each consumer that is incrementally added to the COBWEB tree, a unique identifier for that instance is stored within the red-black tree along with a pointer to its location within the COBWEB tree. The red-black tree can now be searched to locate the instances which have been clustered with a search time of $O(\log n)$.

Note that we separate the characteristics of consumers, used to build the communities, from the SLA they get after the negotiation process. This way, the information added at the end of the negotiation process for each consumer only has a potential impact on the representative SLA configuration for its community, which we call its *stereotype*, keeping the integrity of the COBWEB tree unchanged.

The natural way to define meaningful stereotypes associated to each community is by trying to identify patterns in those service configurations that are representative of the participating consumers and significantly different from the descriptions generated for the other communities. For that, we use a specific metric based on a probability matching scheme [1], since COBWEB's cluster definitions are probabilistic. The metric measures the occurrence increase rate of a specific QoS preference within a given community, when compared to the default occurrence in the whole number of available observations, as an indication of the increase in the predictability of a particular feature (in our case, a given preference) within a community.

Given a QoS preference p, with the default occurrence o_p, if the occurrence rate of this preference within a community i is o_i, the occurrence increase is defined as a simple difference of the squares of the two incidence rates:

$$OI_p = o_i^2 - o_p^2 \ . \tag{1}$$

Clearly, when OI_p is negative, there is a decrease in the occurrence of preference p in that particular community. As such, we can safely conclude that the corresponding preference is not representative of that community. On the other hand, a community's representative characteristic is found through $OI_p > \alpha$, where α is pre-established as the required threshold for considering that occurrence increase enough relevant.

By varying α it is possible to change the proportion of service characteristics covered by the generated stereotypes as well as the amount of overlap between those generated stereotypes. The greater this value, the more dissimilar are the constructed stereotypes and, therefore, the concepts they represent. The impact of adjusting α on the characterisation of the constructed communities will be evaluated in the next section.

3 Evaluation

We are exploiting a set of innovative techniques to develop a cloud marketplace offering personalised multimedia services to end users as a use case. From the end consumer perspective, the developed marketplace is web portal providing a set of personalised services oriented to multimedia enjoyment. The broker interacts with content providers (raw contents) and processing providers (digital multimedia processing), enabling consumers to access (via streaming and download) audio and video content adapted to their QoS preferences (in terms of content features such as speech-to-text, subtitle languages, metadata annotation) and their devices (in terms of content format characteristics such as resolution, encoding, sampling).

To evaluate the effectiveness of the proposed approach, we first applied COBWEB on the task of constructing consumer communities from a set of incremental observations. The dataset for each simulation run contained 1500 consumers. Each consumer and their devices were characterised by a set of 8 attributes. To each attribute, a value from a set of five possible options specific to that particular attribute was randomly chosen according to a Gaussian distribution.

Given this input, COBWEB created a balanced hierarchy whose first three levels are depicted in Figure 1. The numbers in brackets correspond to the size of the corresponding subset of consumers. A first important conclusion about the generated hierarchy is the balanced split of consumers in different communities. Therefore, the underlying concepts are of similar strength.

Then, in order to see the impact of the necessary occurrence increase threshold on the characterisation of these communities, we varied α and measured the

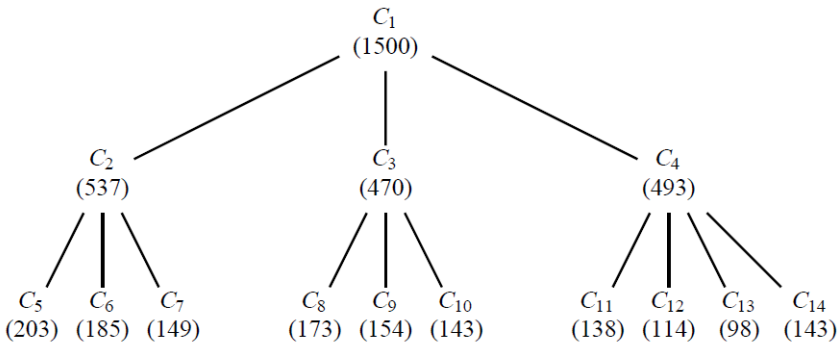


Fig. 1 Hierarchy generated by COBWEB (top three levels)

coverage and overlap of the generated community descriptions. For each consumer, a final SLA was randomly chosen from a set of 50 pre-defined possible service configurations specific to each generated community, simulating the negotiation process. Each SLA was randomly defined from a set of 5 QoS dimensions, 10 attributes for each dimension, and 15 possible values for each attribute. We examined two different partitions of consumers, corresponding to the second and third levels of the generated hierarchy. Figures 2 and 3 reveal the impact of the imposed threshold on the coverage degree in both levels.

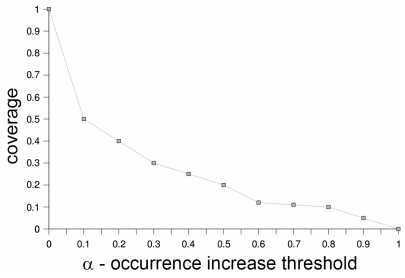


Fig. 2 Coverage degree as a function of α at the second level of the COBWEB tree

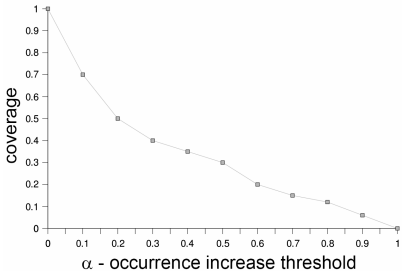


Fig. 3 Coverage degree as a function of α at the third level of the COBWEB tree

As expected, the coverage degree decreases as the criteria for considering a QoS preference as representative of a particular community increases. Furthermore, the coverage on the second level of the hierarchy is consistently lower than that on the third level. This is because the clusters on the third level are more specialised than those on the second. On the other hand, the larger the number of the communities, the larger the overlap between their descriptions. Thus, there is a trade-off between coverage and overlap, as the number of communities increases.

Ideally, we would like to acquire descriptions that are maximally distinct, *i.e.*, minimise the overlap, and increase coverage. As such, the value of α should be relatively small in order to achieve a higher coverage degree. Nevertheless, a value that is excessively small will result in considering that even a small increase in the occurrence of a particular preference to be representative of that community.

Furthermore, the conducted evaluation allowed us to conclude that a large proportion of the considered QoS preferences are not covered in the constructed stereotypes. In general, these are the preferences that are chosen by either too few or too many users. In the former case, the metric ignores them during learning and, in the latter case, they correspond to such general interests, that they cannot be attributed to particular communities. Filtering out these two types of category is a positive feature of the proposed occurrence increase metric.

In a second study, having the set of stereotypes from the previous simulations, we wanted to evaluate how efficiently they would adapt to changes in the SLAs proposed by service providers. In each simulation run, we focused our attention on the stereotype of a particular community, using a value for α of 0.05.

In each simulation run, a set of 1000 new consumers was randomly generated, with at least 40% of them belonging to a subset of the considered community. We evaluated the adaptation ability of the constructed stereotype in two different scenarios. In the first one, the SLAs proposed by the majority of service providers had small changes, either introducing, deleting, or changing a small number of QoS attributes proposed and/or their values. In the second one, more extreme changes that could imply a complete new SLA configuration with new QoS dimensions and attributes from a vast majority of service providers were generated.

In both scenarios, after 5% of new consumers added to any subset of the considered community, the new values or attributes proposed by service providers started to appear in the community's stereotype and were completely replaced around the addition of 20% new consumers. Therefore, we can conclude that the ability of the proposed metric to adapt to changes is relatively fast and correct. Only after a new set of service characteristics is significant in that community can it be considered as part of its stereotype, replacing the old values.

However, it was noticed that in extremes cases, like the one created for the second scenario, old values or attributes that were no longer proposed by service providers were still present in the community's stereotype and, therefore, sent as advice to new consumers. This is not the type of behaviour consumers will expect from a broker. As such, the broker's ability to adapt to changes in a dynamic cloud market was improved by speeding up the remotion of characteristics from the community's stereotype that are consistently no longer proposed by service providers.

In another simulation set, with the same parameters of the previous one, but using the new policy for stereotype adaptation, it was observed that while the values were maintained in the first scenario, it lowered to 2% when service providers no longer propose the service characteristics described in a community's stereotype.

4 Conclusions and Future Work

As the cloud service model matures and becomes ubiquitous, it raises the possibility of improving the way services are provisioned and managed. The main contribution of this paper concerns the design of a broker that can organise consumers into meaningful communities and extract meaningful service configuration stereotypes for each generated community. This allows the broker not only to provide advice and better match each consumer's needs to the available service offers, as well as to help service providers to target service configurations at specific market segments. To the best of our knowledge, no other work introduces the broker as an entity able to understand the needs of customers and providers and capture these needs with clear service stereotypes that focus on the outcome, hiding all details related to service parameters and the low-level infrastructure.

The analysis reported in this paper is just the initial step in the development of an innovative user-centric cloud brokerage service. The next steps will be to continue to develop a novel SLA negotiation algorithm and implement a first real-world prototype. Intensive experiments will exploit this first prototype and provide valuable feedback to the research line that we intend to follow.

Acknowledgements. This work was partially supported by LIACC through Programa de Financiamento Plurianual of FCT (Portuguese Foundation for Science and Technology).

References

1. Biswas, G., Weinberg, J., Fisher, D.: Iterate: a conceptual clustering algorithm for data mining. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 28(2), 219–230 (1998)
2. Cuomo, A., Modica, G., Distefano, S., Puliafito, A., Rak, M., Tomarchio, O., Venticinque, S., Villano, U.: An sla-based broker for cloud infrastructures. Journal of Grid Computing 11(1), 1–25 (2013)
3. Fisher, D.H.: Knowledge acquisition via incremental conceptual clustering. Machine Learning 2, 139–172 (1987)
4. Gluck, M., Corter, J.: Information, uncertainty and the utility of categories. In: Proceedings of the 7th Conference of the Cognitive Science Society, pp. 283–287 (1985)
5. Guibas, L.J., Sedgewick, R.: A dichromatic framework for balanced trees. In: Proceedings of the 19th Annual Symposium on Foundations of Computer Science, pp. 8–21 (1978)
6. Kyriazis, D.: Cloud computing service level agreements - exploitation of research results. Tech. rep., European Commision (2013)
7. Nair, S., Porwal, S., Dimitrakos, T., Ferrer, A., Tordsson, J., Sharif, T., Sheridan, C., Rajarajan, M., Khan, A.: Towards secure cloud bursting, brokerage and aggregation. In: 8th IEEE European Conference on Web Services, pp. 189–196 (2010)
8. Paliouras, G.: Discovery of web user communities and their role in personalization. User Modeling and User-Adapted Interaction 22(1-2), 151–175 (2012)
9. Tordsson, J., Montero, R.S., Moreno-Vozmediano, R., Llorente, I.M.: Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers. Future Generation Computer Systems 28(2), 358–367 (2012)

Multi-agent Negotiation of Decentralized Energy Production in Smart Micro-grid

Alba Amato, Beniamino Di Martino, Marco Scialdone, and Salvatore Venticinque

Abstract. SmartGrid is an electricity network that can intelligently integrate the actions of all users connected to it in order to efficiently deliver sustainable, economic and secure electricity supplies. In this context the CoSSMic project aims at fostering a higher rate of self-consumption of decentralized renewable energy production, using innovative autonomic systems for management and control of power micro-grids on users behalf. To achieve this goal we have designed an ICT framework that integrates different appliances such as smart meters, solar panels, batteries, etc., providing a common platform to support sharing of information and negotiation of energy exchanges between power producers and storages in accordance with policies defined by owners, weather forecasts, and habits and plans of participants.

Keywords: Multi-Agent Systems, Smart Grid, Energy Market, XMPP.

1 Introduction

A SmartGrid is an electricity network that employs innovative products and services together with intelligent monitoring, control, communication, and self-healing technologies to better facilitate the connection and operation of generators of all sizes and technologies. It allows consumers to play a part in optimizing the operation of the system. It significantly reduces the environmental impact of the whole electricity supply system. CoSSMic (Collaborating Smart Solar-powered Micro-grids - FP7-SMARTCITIES-2013) is an ICT European project that aims at fostering a higher rate for self-consumption (<50%) of decentralized renewable energy production by

Alba Amato · Beniamino Di Martino · Marco Scialdone · Salvatore Venticinque
Department of Industrial and Information Engineering,
Second University of Naples, Via Roma 29, Aversa, Italy
e-mail: {alba.amato,beniamino.dimartino,
marco.scialdone,salvatore.venticinque}@unina2.it

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_17

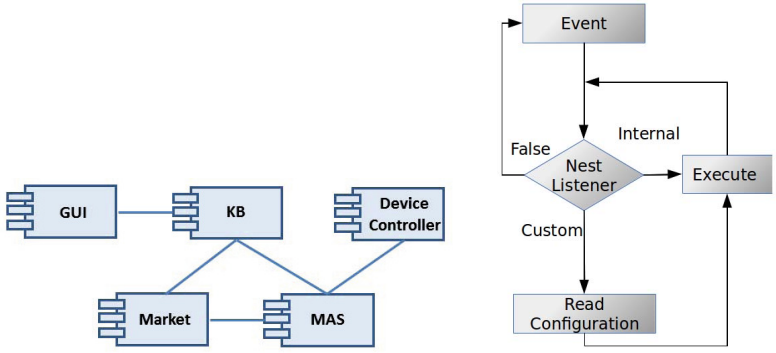
innovative autonomic systems for management and control of power micro-grids on users' behalf. This will allow power optimization of household and neighborhood and of sales to the network. In addition CoSSMic will provide a higher degree of predictability of power deliveries for the large power companies, and it will satisfy the requirements and achieve the benefits discussed above. To obtain a more profitable distribution of energy for utilities and customers it is necessary a system able to trade energy allocation and price information. Besides a 'smart' system has to manage the information's exchange within the network, the integration of renewable energies, the strategy to reduce costs or for optimization of energy consumption and distribution. To obtain these goals we propose a modular, vendor agnostic, agent based architecture that uses the publish/subscribe paradigm. Cloud services will be provided to connect distributed installations, to allow for power monitoring and updating policies by users. The agent based approach was chosen to run each software instance autonomously. The design and development of such a framework should support the communication among agents over a peer-to-peer overlay to negotiate the scheduling of power sources to energy storages. In this paper we address the problem of designing a framework that supports agents based P2P negotiation among power consumers and producers for the optimal scheduling of energy exchange.

2 Related Work

The scientific community investigates different priorities in the field of smart grids. Much effort has been spent on the investigation in this field of agents technology [7]. In [6] authors consider how consumers might relate to future smart energy grids, and how exploiting software agents to help users in engaging with complex energy infrastructures. In [1] authors describe a Message Oriented Middleware (MOM) to simplify distributing applications across heterogeneous operating systems, programming language, computer architectures, networking protocols, and at the same time reducing the complexity on the interconnection functionalities and providing a high level of scalability based on RabbitMQ, Data Distribution Service (DDS) and the Extensible Messaging and Presence Protocol (XMPP). In [5] multi-agent resource allocation in a competitive peer-to-peer environment is addressed making use of micro-payment techniques, along with concepts from random graph theory and game theory. It provides an analytical characterization of protocol and specifies how an agent should choose optimal values for the protocol parameters. In [4] authors claim that agent and peer-to-peer based decentralized self-management can change the future of energy markets in which the power grid plays a core role. Our contribution, and in particular the CoSSMic project, supports negotiation among end users on real power grid.The framework will be validated on real infrastructures by trials that involve inhabitants of two different European countries. Both software and hardware will be integrated and customized to be compliant with existing installations. Finally we have experience in building network of agents both in smart cities applications [2] and for negotiation and brokering of computational resources in Cloud markets [3].

3 Architecture

The stakeholders of CoSSMic Framework include Users, Devices and Power Suppliers (GenCO). The CoSSMic User interacts with the framework by a Graphical User Interface (GUI) according to three high level use cases (UCs). The *Management* UC allows to configure and manage the available devices and to manage and control at a higher level, through rules and policies, the energy flows. The *Monitoring* UC provides facilities to supervise and to get useful information for eventually reconfiguring the devices and scheduling the allocation of power. *Reporting and statistics* integrate information from several sources, including power companies, weather reporting and forecasting and to encourage the growth of the neighborhood network. CoSSMic Devices will use the Platform providing metering and management services. CoSSMic will exchange power with GenCO when the MicroGrid cannot satisfy its requirements in the case of over- or under- production of energy within the CoSSMic neighborhoods. The CoSSMic platform will run on embedded computer systems that will be provided to end users as a black box, to be plugged into the power network and connected to Internet. The Platform will be installed in every household and will join a community of other instances within the neighborhood. Instances of the platform communicate by a P2P overlay and with the Cloud to eventually exploit advanced services. Each platform instance will communicate with other households only for the energy negotiation. Components of the CoSSMic Platform are shown in Figure 1 (a).



(a) CoSSMic Platform (b) Control Agent Life Cycle

Fig. 1 Control Agent Life Cycle and Contract Net Interaction Protocol

A Graphical User Interface (GUI) allows users to interact with electronic devices through graphical icons and visual indicators. The Knowledge Base (KB) is an information repository that provides means for information to be collected, organized, shared, searched and utilized. The Multi Agent System (MAS) allows for the deployment of agents of consumers and producers that will participate in the energy distribution. The Market supports the energy negotiation. Agents can publish calls

for proposals, accept offers and negotiate with other agents. The Device Controller allows agents to send real time commands to electric devices in the smart house. Agents will act on user behalf and their main task is to negotiate energy. They can be classified according to two categories. *Consumers* buy energy for passive devices. *Producers* can sell energy. In this category there are, for example, power generators, solar panels, wind turbines, but in CoSSMic we only consider solar panels. Those devices, which are able both to produce and consume energy will be defined *Prosumers*. In this category there will be also storages, which are represented by a couple of agents belonging to the two different classes.

The consumer agents, once found one who offers enough power to meet their own needs, will kick off the negotiation. If an agent cannot find enough energy to satisfy its needs in the neighborhood market, it will contact a GenCo.

4 Implementation

To develop and deploy agents, we used the JADE multi-agent system[1]. Figure 2 shows the implementation of our agents-based architecture. The GUI, described previously, allows the user to interface with the system. The various devices in the home can send information about electricity consumption through wireless interfaces (for example UHF or Zigbee) to the mediator. Mobile devices (e.g. electric cars), instead, send information through the CoSSMic Cloud. In both cases the information, through the Mediator, reach the agent platform whose main actors are: *User Agent*: an agent associated to the user that interfaces with the GUI and with the DB Manager; *Event Bus*: handles the various possible events. Control Agents subscribe to this event bus in order to receive events from devices; *DB Manager*: interfaces with the Knowledge Base that, as we said, can collect, manage and share information; *Control Agent*: the most important of all, it manages the electric energy of all devices in the household. In particular, in our MAS we have two types of control agent. A *Consumer Agent* is associated to each electric device that absorbs electricity. Its task is to obtain the energy required from the device to operate. A *Producer Agent* is associated to each electric device that produces electricity and it tries to sell this energy to consumer agents. In Figure 1 (b) the control agent life cycle is shown. When an event occurs, the CA calls all the registered listeners. Some event and listeners are built in, but the developer is able to define new custom events and custom listeners.

A configuration file is used to define those events, and related listeners, which the agent must react to. In our prototype we already provide the Negotiation Handler as a listener that can be used to start a new negotiation on a specific event. In the configuration itself the developer can choose what type of negotiation protocol to be used and the related negotiation strategy. We already support a negotiation protocol based on the FIPA Contract Net Interaction Protocol. In our test, the initiator starts the negotiation when the listener receives an event indicating that the

[1] `http://jade.tilab.com/`

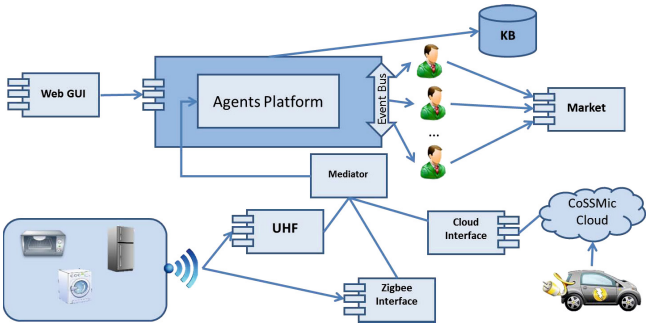


Fig. 2 Implementation of CoSSMic Platform

energy level is below a certain threshold and the initiator sends a CFP to a number of power producers who are already running. At this point there are two possible situations. The responders send a message with the energy price, the initiator accepts one proposal and rejects all the others, or the initiator rejects all the proposals or more simply there are not proposals. In the last case the agent changes its own behaviour from initiator to responder, waiting for call for proposal by new producers. A straightforward negotiation strategy could be the acceptance of the proposal that offers the lower price among the ones which are below a thresholds. Of course different strategies can be defined and configured by the developer.

At transport layer to support P2P negotiation protocols we use the eXtensible Messaging and Presence Protocol (XMPP)[2]. Our solution involves the use of a chat room that acts as a market for a neighborhood where agents enter and publish their requests for sale or purchasing energy. The prototype is implemented using TIGASE, an open source project to develop a XMPP server implementation in Java. Furthermore, to realize the communication between agents through XMPP, we use Smack[3], an Open Source XMPP (Jabber) client library for instant messaging and presence. Using an XMPP server, it is possible to create rooms - open or reserved - where users can meet and communicate. In the implementation of the prototype, we use a persistent Chat Room (in this way, if everyone leaves the chat or if the server restarts, the room remains and stores all messages sent) as Energy Market, in which the control agents can enter autonomously. Once entered the market room, producer agents can make proposals. If a consumer agent believes that an offer is satisfactory, it starts a negotiation with the agent who made the proposal.

[2] http://www.xmpp.org

[3] http://www.igniterealtime.org/projects/smack/

5 Conclusion

The paper presents a modular software architecture for supporting agents to collect information about local energy production and storage resources of neighborhoods of individual houses and to schedule the energy flows using negotiation protocols. Besides the implementation of a preliminary prototype is described. The introduction of the publish/subscribe paradigm as a solution for service data exchange within a smart grid was introduced together with the usage of the XMPP protocol to allow secure interaction with other systems allowing the extension of the architecture to other platforms with little effort. Experimental activities are an ongoing work together with the implementation of optimal negotiation strategies and new protocols. We also plan, as a future work, to use a NoSQL database for metering and monitoring all smart micro grids participating in the trials of the project.

Acknowledgements. This work has been supported by CoSSMic (Collaborating Smart Solar-powered Micro-grids - FP7-SMARTCITIES-2013).

References

1. Alkhawaja, R.A., Ferreira, L.L., Albano, M., Garibay-Martínez, R.: Qos-enabled middleware for smart grids. Tech. rep., Polytechnic Institute of Porto, ISEP-IPP (2012)
2. Amato, A., Di Martino, B., Venticinque, S.: Semantically augmented exploitation of pervasive environments by intelligent agents. In: Proceedings of the 2012 10th IEEE International Symposium on Parallel and Distributed Processing with Applications, ISPA 2012, pp. 807–814 (2012)
3. Amato, A., Liccardo, L., Rak, M., Venticinque, S.: Sla negotiation and brokering for sky computing. In: Proceedings of the 2nd International Conference on Cloud Computing and Services Science, CLOSER 2012, pp. 611–620 (2012)
4. Brazier, F., Ogston, E., Warnier, M.: The future of energy markets and the challenge of decentralized self-management. In: Beneventano, D., Despotovic, Z., Guerra, F., Joseph, S., Moro, G., de Pinninck, A.P. (eds.) AP2PC 2008/2009. LNCS (LNAI), vol. 6573, pp. 95–103. Springer, Heidelberg (2012)
5. Peleg, Y., Rosenschein, J.S.: Agents and peer-to-peer computing: Towards P2P-based resource allocation in competitive environments. In: Beneventano, D., Despotovic, Z., Guerra, F., Joseph, S., Moro, G., de Pinninck, A.P. (eds.) AP2PC 2008/2009. LNCS (LNAI), vol. 6573, pp. 129–140. Springer, Heidelberg (2012)
6. Rodden, T.A., Fischer, J.E., Pantidi, N., Bachour, K., Moran, S.: At home with agents: Exploring attitudes towards future smart energy infrastructures. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2013, pp. 1173–1182. ACM, New York (2013)
7. Rogers, A., Ramchurn, S.D., Jennings, N.R.: Delivering the smart grid: Challenges for autonomous agents and multi-agent systems research. In: AAAI (2012)

Towards Elastic Component-Based Cloud Applications

Alexander Pokahr and Lars Braubach

Abstract. In the context of cloud computing, elasticity refers to the capability of an application to automatically adapt its resource consumption to the current demand, i.e. scale out or scale in as the load increases or decreases. Today's platform-as-a-service (PaaS) cloud solutions are strongly biased towards typical client/server, web-based applications. The provided runtime environments and services are thus only of limited help for building traditional business applications. In this paper, a vision of elastic component-based applications is presented that is aimed at supporting elasticity for arbitrary application designs. This vision is supported by a component-based programming model for allowing developers to specify application functionality in a cloud-enabled way and a runtime environment that supports the specification and automatic management of non-functional attributes, such as application response times. For both areas, the programming model as well as the management of non-functional attributes, the paper presents the challenges, an approach for realizing the ultimate vision, as well as our initial results.

1 Introduction

Cloud technology promises the unlimited availability of computing resources based on pay-per-use accounting models [2]. By only consuming as much resources needed at any point in time, applications are enabled to handle arbitrary system loads at the lowest possible cost. This adaptive property of cloud-hosted applications is called elasticity [11]. The two main advantages of elasticity are on the one hand the ability to handle in principle any number of users simultaneously without noticing any degradation of performance and on the other hand the efficient use of resources, i.e. only having to allocate computing resources that are actually needed. Cloud

Alexander Pokahr · Lars Braubach
Distributed Systems and Information Systems Group, University of Hamburg, Germany
e-mail: {pokahr,braubach}@informatik.uni-hamburg.de

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_18

platforms following the platform-as-a-service (PaaS) model offer programming environments, where elastic applications can be deployed. PaaS platforms simplify developing elastic applications by letting developers concentrate on application functionality.

Today's PaaS infrastructures, such as Google App Engine,[1] are targeted towards "web-accessible applications" [15] that have to process many, yet typically simple requests. E.g., Google App Engine requires requests to be served in around 60 seconds. These PaaS infrastructures offer runtime environments, e.g. for Java Servlet-based web applications, and commonly required services, such as user authentication and data stores. When a system deviates from the web-accessible application class, elasticity is not easily achieved [22], because applications need to be carefully designed to avoid any form of bottlenecks that would incur inacceptable overheads for doing a scale out. An example is a complex calculation algorithm. If it is deployed as is in the cloud, there is no elasticity regarding the size of the calculation problem, as according to Amdahl's law [1] the computation speed cannot be increased by scaling out. The developer would have to split the computation manually into parallelizable chunks, e.g. by applying suitable algorithmic skeletons [10] such as map-reduce.

This paper presents a vision of elastic component-based applications. The goal is to support the development of arbitrary component-based applications by empowering developers to consider application elasticity with regard to explicit non-functional requirements seamlessly during design, implementation and operation of applications. This paper is structured as follows. In Section 2, the vision of elastic component-based applications and the chosen approach are presented. As two fundamental building blocks of the approach, Section 3 introduces a generic component model for elastic application programming and Section 4 presents adaptation management for elastic applications. Afterwards, Section 5 discusses related work and section 6 finally concludes the paper.

2 Vision

There is no common definition of the term elasticity in the context of cloud computing [11], but there seems to be some agreement that elasticity implies a (typically automatic) adaptation of a system to match its current workload, i.e. to allocate more resources when the workload increases and to de-allocate resources when they are not needed. Defining elasticity in terms of a match between workload and resources leads to the questions, how resources and workload are quantified and how the suitability of a match is determined. For web-accessible applications, these questions are easily answered. *Resource quantity* is given by the allocated instances of services provided by the PaaS infrastructure, such as the number of front end instances in Google App Engine. *Workload quantity* is given by the number of requests to a service. The *suitability of the match* follows from the common requirement of all

[1] `https://cloud.google.com/products/app-engine/`

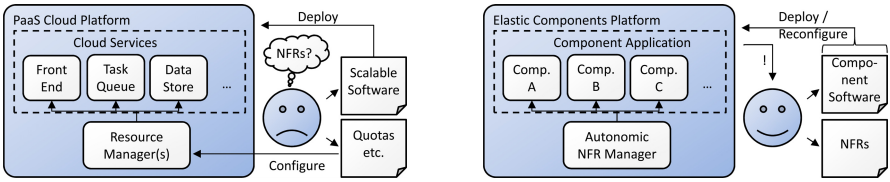


Fig. 1 Current state (left) and vision (right)

interactive applications to be responsive to user requests, i.e. allocate resources as needed to service any request in less than 10 seconds in order to keep the users attention [17].

These answers certainly do not hold for arbitrary applications. The granularity of the resource quantity of an application should not be restricted to services provided at the platform level, but rather allow scaling at the level of individual application components, defined by the developer. The structure of workload and its respective quantity is highly application dependent and will be composed of a spectrum of high or low numbers of simple or complex computations inside the components. Finally, the suitability of the match is defined in terms of non-functional requirements (NFRs) of the application and its individual functionalities. E.g., for background processing, a response time much larger than 10 seconds would still be acceptable, while for real-time control, response times should be in the range of milliseconds. In addition to response times, other NFRs might apply to the application, such as security (e.g. data that should not be transferred across data centers) or safety (e.g. requiring a level of redundancy for important subsystems).

We envision a programming model and runtime environment, which aids developers in building arbitrary applications exhibiting the notion of elasticity defined above. The current state and the vision are depicted in Fig. 1. The left hand side shows, that a developer currently has to design scalabe software by hand using predefined cloud services. By configuring runtime parameters, such as quotas, the developer only has implicit control over how the deployed application meets the NFRs. In the envisioned approach (right hand side), the developer uses a programming model that allows specifying application functionality in a clean component-based design and that supports the direct specification of NFRs as part of or separate to the application model. The software and its NFRs can be deployed together to a runtime infrastructure, which is responsible for managing NFRs. Using explicit NFR knowledge, an autonomic manager is in charge of dynamically adapting the application according to elasticity dimensions and inform the developer at any time how the NFRs are met.

3 Elastic Component Model

In an ideal world, the programmer would only need to concentrate on functional application logic and the infrastructure would know how to partition an application into suitable parallelizable and distributable chunks to meet every possible require-

ment with regard to elasticity. In the real world, this partitioning has to be performed manually and the developer has to determine herself, by performing a mixture of complex upfront analysis and trial and error, if a certain application design meets certain elasticity requirements.

A generic elastic application programming approach should empower the developer to consider elasticity during design and runtime of the application. This would require a language to specify application functionality in combination with meta-information about how to partition the application without endangering consistency or degrading performance. During the design phase, tool support could be provided for generating design critics, e.g. by identifying potential bottlenecks and during runtime, the infrastructure would know how to partition the application into parts that can be allocated to different nodes.

3.1 Challenges

Distribution Transparency: For simplicity, the programming model would exhibit a high degree of distribution transparency, i.e. the developer would be relieved from issues of remote communication, resource discovery, parallel execution and synchronization. This leads to application designs that are easy to understand, but with poor performance, if parts of the application are distributed.[2] The opposite extreme would be programming for the most complex case, i.e. explicitly as distributed and parallel as possible. An elastic programming model has to find a good trade-off between these extremes, i.e. being simple while still promoting scalable application designs.

Execution Environment: Distribution transparency requires that the infrastructure provides capabilities that are hidden from the developer. To support elasticity, at least replication, distributed allocation, binding and communication need to be managed by the infrastructure. To react to changing application load, mechanisms need to be provided for replicating application parts and allocating replicated parts to remote nodes. Once application parts are transparently allocated to distributed nodes, the infrastructure also needs to provide transparent binding, e.g., using load balancing strategies, and transparent remote communication. For determining the fulfillment of NFRs at runtime, the execution environment furthermore needs mechanisms for instrumenting deployed applications in order to measure values for non-functional properties.

[2] Cf. Fowler's *First Law of Distributed Object Design: Don't distribute your objects!* [9].

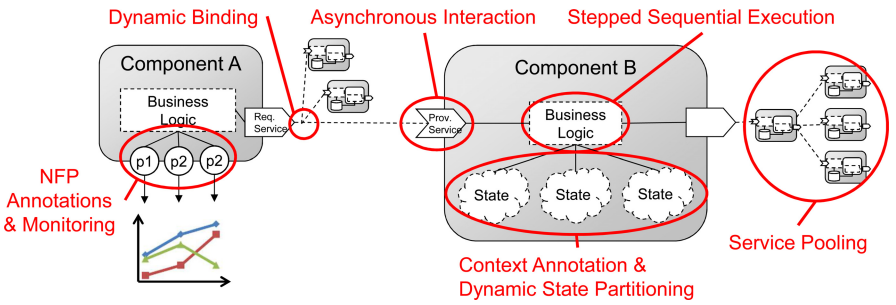


Fig. 2 Elastic components approach

State Management: Scaling mechanisms need to support consistency of runtime state when replicating application parts. E.g., in web-accessible applications, session data is associated to ongoing user interactions. Web sessions are independent of each other making state management comparably simple. In complex business applications, the interdependencies between states of application parts might not be obvious and thus it is unclear, which kind of consistency is required. Relaxing consistency (e.g. BASE instead of ACID) can improve scalability [21]. Thus the programming model should enable the developer to specify the required level of consistency for dependent states and also allow the runtime environment to determine, which application states are independent and thus do not need consistency at all.

3.2 Approach

To meet the challenges described above, a so called *elastic component model* is proposed (cf. Fig. 2). This model is based on an existing component model for distributed computing called active components [19], and extends it with features for dealing with elasticity. Active components are defined in terms of required and provided services as well as internal business logic. They only interact through the required and provided service interfaces and the binding between components may be performed dynamically at runtime. Similar to the actor model [12], active components are executed internally in small sequential steps and interaction between components is always asynchronous. The active components approach provides a good trade-off for the distribution transparency challenge, as it allows abstracting away from tedious low-level details, such as service binding and network communication. Asynchronous service interfaces guide the developer towards scalable application decompositions, i.e. partitioning the business logic into synchronous computation, which should be performed locally inside a component, and asynchronous interaction, that can be distributed without severe performance loss.

Current Implementation: Active components are realized in the open source Jadex platform,[3] which provides a programming API, tools and an execution environment for active components. During design and implementation, required and provided services of components can be modelled as Java interfaces. As a first step towards extending active components to support elasticity, NFP-related meta-information can be attached to the interfaces (e.g. method signatures) and implementations of the services using Java annotations. The Jadex runtime supports monitoring of these NFPs and provides service pools, for load balancing and instance management, i.e. creating or destroying component instances as a reaction to NFP changes. Currently, Jadex does not support fully transparent replication of components. Service pools have to be an explicit part of the design and can only be applied to stateless components or components that use a separate context service for state management.

Outlook: The elastic component model should also allow transparent replication of any application component. This can only be achieved, when all component state is managed by the infrastructure, such that replication can automatically respect constraints with regard to state consistency. We envision adopting a model similar to sessions in web applications, that allows to partition the state of a component into independent contexts. In arbitrary component-based applications, the definition of context is more complex than for web applications. Therefore, the developer of the application should be enabled to annotate context information to state (e.g. class variables in a Java component implementation) and computation (Java method implementations). These annotations will allow intermediate forms between fully stateful and fully stateless components: elastic components with their state dynamically partitioned into sets of contexts that may have relaxed or no consistency requirements between them.

4 Goal-Based Adaptation Managment

An implicit part of a cloud PaaS, which is not perceived by a cloud user, is its application adaptation management. Despite its invisibility, it is at the heart of the PaaS infrastructure and responsible for achieving the elasticity of an application. In case of traditional web-based applications, adaptation management is a rather straightforward task as elasticity is achieved exclusively by distributing web requests to different servers and starting/stopping new VMs if needed. This kind of adaptation management is well-known from large-scale web server architectures relying on load balancers. These load balancers are in charge of distributing requests according to different algorithms such as round-robin, LRU etc. If the application model is extended towards general component-based applications these simple adaptation strategies do not suffice any longer and the adaptation manager is equipped with an extended set of adaptation mechanisms. In addition to infrastructure operations such as adding or removing VMs, also operations at the component level need to be

[3] `http://www.activecomponents.org/`

considered. These may include replication or migration of components to adapt to demand changes.

From a user perspective, current adaptation management solutions are limited with respect to the way the user objectives and expectations of an application can be specified. Solutions like the Google App Engine only allow for defining very technical settings of the infrastructure including, e.g., the workload and maximum idle times triggering the creation and shutdown of new VM instances. Moving to the general case of component applications it makes sense to allow for specifying NFRs instead of technical elasticity settings, because they describe what a user expects from the application. Having acquired the non-functional expectations of an application, the adaptation manager has to perform adaptation actions to ensure their continuous fulfillment. In this sense, the adaptation manager becomes an autonomic system monitoring the NFPs of the application, using them to compute NFR violations and based on violations is in charge of initiating adaptation procedures.

4.1 Challenges

NFR Descriptions: A basic challenge consists in the description of non-functional requirements of an application. Typically, NFRs are domain-dependent as well as domain-independent rendering the task of requirement descriptions quite challenging. Furthermore, the dependencies between NFRs are of importance, because it might often be the case that conflicts between requirements exist so that they cannot be fulfilled simultaneously. In such cases the adaptation manager needs hints which requirement is more important than another in order to prioritize adaptation procedures.

Mechanism for Explaining Adaptations: In case of component-based systems the available adaptation mechanisms become more diverse. If some of the NFRs are not satisfied during runtime or the system configuration is unintuitive for the administrator of the application, the system should be able to explain the reasons for its adaptation actions. This means that an administrator should be enabled to see which adaptations have been performed and ask why certain of those adaptations have been performed.

Execution of Policies: Once the policy model has been derived, it is subject to continuous runtime execution. A policy engine has to ensure that the internal NFR representation is updated with regard to NFP changes in the monitored application. Given that NFRs become unsatisfied the engine has to deliberate between the achievement of different NFRs and select adaptation procedures.

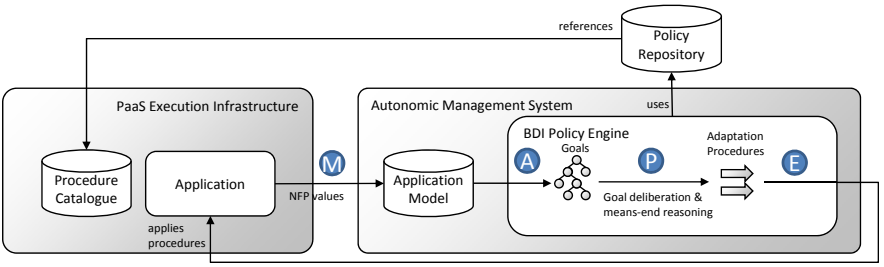


Fig. 3 Autonomic PaaS Cloud Manager

4.2 Approach

The challenges from above will be answered with an architecture proposal for an autonomic PaaS manager (cf. Fig. 3) realizing the MAPE-loop [14]. The underlying idea is that an application has to provide non-functional properties that can be measured at runtime and which will be used to validate if and to what degree the user defined NFRs are satisfied. In case of violations the application is modified by applying adaptation procedures.

In order to describe NFRs of an application, a goal-based approach similar to Tropos [5] is used, that allows users to express their requirements in terms of a hierarchy of soft-goals. The NFR descriptions are constrained by two aspects. First, the NFRs can only be selected according to a predefined set of NFR types (e.g. minimize response time or maintain CPU usage below 80%) and second, NFRs need to be connected to NFPs of the application in order to become automatically evaluable at runtime. In addition to the NFRs, also their interdependencies need to be modelled by stating which NFRs are more important than others and which ones may conflict.

Current Implementation: Internally, the execution of policies requires an engine that explicitly handles goals and realizes a practical reasoning process determining the actions to be taken. We propose to follow the BDI (belief-desire-intention) agent approach [4], which is based on two subprocesses: a) goal deliberation and b) means-end reasoning. The first is responsible for prioritizing goals and finding a conflict-free set to pursue and the latter is responsible for selecting among possible adaptation procedures depending on the current context. The practical reasoning process has been implemented as part of the Jadex BDI agent framework [20], which forms the base of the novel policy engine.

Outlook: In a first evolution of the system, adaptation policies will have to be modelled explicitly by the user as part of the policy description. For this purpose the user can choose among predefined actions from a procedure catalogue. The actions are rather generic and belong either to the VM infrastructure or to the component level. On both layers new elements can be created or deleted and on the component layer, elements can also be replicated and migrated. In further evolutions of the sys-

Approach	*Dis. Transparency*	*Exe. Infrasturcture*	*State Management*	*NFR Descriptions*	*Explanations*	*Policy Execution*
CCM	async. RMI	REST, message queues, databases	stateless components	NFPs	n/a	n/a
COSCA	RMI, service registry	distributed OSGi	stateless components	NFPs	n/a	n/a
Aneka	RMI, node integration	.NET	bag of tasks, workflows, map-reduce	QoS/SLAs	n/a	workload scheduler, resource reservations
mOSAIC	drivers, connectors	virtual machine	stateless components	SLAs	n/a	agent-based (re)config.
Paremus	SCA	distributed OSGi	stateless components	NFPs	n/a	process groups
EventWave	event handler	Mace	contexts	n/a	n/a	n/a
SYBL	n/a	n/a	n/a	NFRs and NFPs	n/a	distributed services

Fig. 4 Overview of alternative approaches

tem it will be investigated if an automatic NFR to policy conversion can be done, relieving developers from the tedious task of selecting adaptation actions. The main advantage of using a goal-based approach is that the system can be exploited to give rational explanations for adaptation procedures that have been applied, because the justification for actions is known. As a prerequisite for an explanation system the policy engine has to write logs with the current goal hierarchy whenever adaptation procedures are executed. Based on these logs, relatively simple algorithms [6] can be applied to generate human understandable explanations.

5 Related Work

The idea of extending cloud PaaS towards the component-based software engineering paradigm has already been tackled by some approaches including the cloud component model (CCM) [3], COSCA [13], mOSAIC [18] and Paremus [16] in the context of grid systems. In addition we have also considered approaches that aim at NFR enforcement in cloud systems, which is tackled by Aneka [23], EventWave [7] and SYBL [8]. To compare those approaches with our proposal we have analyzed in how far they address the identified challenges. The summarized results of this analysis are depicted in Fig. 4.

It can be observed that currently approaches focus on either the programming model or the NFR enforcement (left vs. right side in the figure) but with an exception of Aneka, few approaches already try to combine both. The component-based cloud solutions have in common that all of them establish distribution transparency in a service oriented manner relying on RMI. Paremus uses the service component architecture, while CCM employs RESTful web service and COSCA and Aneka use language specific RMI solutions. EventWave and mOSAIC are event based programming environments in which event handlers are used to encapsulate functionality with communication mechanisms transparently implemented in the middleware.

The underlying component model and especially the handling of state are different in the approaches. COSCA and Paremus are centered on the OSGi component model, Aneka is based on Microsoft's .net and CCM defines its own component model. The OSGi approach introduces bundles as deployment unit and services as communication channel between components. A registry is used to find required services. COSCA as well as Paremus rely on a distributed OSGi version in which service discovery is possible also within a network of nodes. The handling of state

within the OSGi model has not been changed, i.e. components are either stateful or stateless and only the latter ones can be used for scaling. The CCM model defines a component with a functional and a performance interface and only permits stateless components. All state is assumed to be saved in an external database. Aneka uses typical grid programming models like 'bag of tasks' in order to avoid problems with state. With regard to state management, EventWave is the most advanced approach. Similar to our vision, EventWave allows context information to be annotated to the application code and performs dynamic partitioning at runtime.

NFR descriptions are used only in context of Aneka and SYBL. The other approaches are limited towards monitoring with NFPs but without a connection to NFR violations. In Aneka a bag of tasks scheduler has been proposed which is capable to fulfill user defined QoS criteria via exclusive resource reservation and assignment of tasks to suitable resources. In SYBL the whole language has been designed with a focus on NFR violation detection. It allows for specifying NFRs and monitored NFPs and proposes a distributed execution runtime including a coordination service. The technically weak point of the approach is that it has to be connected to the underlying cloud environment and can only utilize monitoring data that is available via the corresponding cloud API.

In summary it can be stated that the challenges of enhanced cloud programming models and NFR monitoring have already been identified, but component models have largely ignored the state problem until now and NFRs have been considered merely on the system level without component connections.

6 Conclusions and Outlook

In this paper a roadmap and achievements towards the vision of an elastic component model for cloud PaaS have been put forward. Such a component model will achieve two things. First, it will allow for making arbitrary applications elastic - in contrast to a web centered approach today - and second, it will help users in operating cloud applications by using NFR monitoring - which is domain oriented and not technically focused as nowadays. We have presented the main challenges in both directions and also provided a solution path towards this vision. In the area of the programming model especially the state problem has to be addressed because current component models only support stateful vs. stateless components and elasticity demands stateless components. To solve this problem a novel context based approach has been sketched which generalizes the concept of web sessions for components. In the area of NFR monitoring the explanation of adaptation actions has been identified as important building block towards comprehensible autonomic behavior from a user perspective. Explanations require that the reasons for applying adaptation procedures can be deduced. In order to achieve this we propose using goal-based NFR descriptions which should also be directly executed with the autonomic manager. As part of future work we continue working on both research strands and plan to conceive a generic context mechanism as well as an execution machinery for our soft-goal based NFR descriptions as next important steps.

References

1. Amdahl, G.: Validity of the single processor approach to achieving large scale computing capabilities. In: Proc. of the Spring Joint Computer Conf., AFIPS, pp. 483–485. ACM, New York (1967)
2. Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: A view of cloud computing. Commun. ACM 53(4), 50–58 (2010)
3. Bahga, A., Madisetti, V.: Rapid prototyping of multitier cloud-based services and systems. Computer 46(11), 76–83 (2013)
4. Bratman, M.: Intention, Plans, and Practical Reason. Harvard University Press (1987)
5. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J.: Tropos: An agent-oriented software development methodology. Autonomous Agents and Multi-Agent Systems 8(3), 203–236 (2004)
6. Broekens, J., Harbers, M., Hindriks, K., van den Bosch, K., Jonker, C., Meyer, J.-J.: Do you get it? User-evaluated explainable BDI agents. In: Dix, J., Witteveen, C. (eds.) MATES 2010. LNCS, vol. 6251, pp. 28–39. Springer, Heidelberg (2010)
7. Chuang, W.-C., Sang, B., Yoo, S., Gu, R., Kulkarni, M., Killian, C.: Eventwave: Programming model and runtime support for tightly-coupled elastic cloud applications. In: Ann. Symp. on Cloud Computing, SOCC, pp. 21:1–21:16. ACM, New York (2013)
8. Copil, G., Moldovan, D., Truong, H.-L., Dustdar, S.: Sybl: An extensible language for controlling elasticity in cloud applications. In: Int. Symp. on Cluster, Cloud and Grid Computing (CCGrid), pp. 112–119 (May 2013)
9. Fowler, M.: Patterns of Enterprise Application Architecture. Addison-Wesley Longman Publishing Co., Inc., Boston (2002)
10. González-Vélez, H., Leyton, M.: A survey of algorithmic skeleton frameworks: high-level structured parallel programming enablers. Softw., Pract. Exper. 40(12), 1135–1160 (2010)
11. Herbst, N., Kounev, S., Reussner, R.: Elasticity in cloud computing: What it is, and what it is not. In: Int. Conf. on Autonomic Computing, pp. 23–27. USENIX (2013)
12. Hewitt, C., Bishop, P., Steiger, R.: A universal modular actor formalism for artificial intelligence. In: Int. Joint Conf. on Artificial Intelligence, IJCAI, pp. 235–245. Morgan Kaufmann Publishers Inc., San Francisco (1973)
13. Kächele, S., Hauck, F.J.: Component-based scalability for cloud applications. In: Int. WS. on Cloud Data and Platforms, CloudDP, pp. 19–24. ACM, New York (2013)
14. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. IEEE Computer 36(1), 41–50 (2003)
15. Krintz, C.: The appscale cloud platform: Enabling portable, scalable web application deployment. IEEE Internet Computing 17(2), 72–75 (2013)
16. Paremus Ltd. The paremus service fabric - a technical overview. Tech. report, Paremus Ltd. (2009)
17. Miller, R.: Response time in man-computer conversational transactions. In: Fall Joint Computer Conf., AFIPS, pp. 267–277. ACM, New York (1968)
18. Petcu, D., Şandru, C.: Towards component-based software engineering of cloud applications. In: Proc. of WICSA/ECSA 2012, pp. 80–81. ACM (2012)
19. Pokahr, A., Braubach, L.: The active components approach for distributed systems development. Int. J. of Parallel, Emergent and Dist. Systems 28(4), 321–369 (2013)
20. Pokahr, A., Braubach, L., Jander, K.: The jadex project: Programming model. In: Ganzha, M., Jain, L.C. (eds.) Multiagent Systems & Applications. ISRL, vol. 45, pp. 21–53. Springer, Heidelberg (2013)
21. Pritchett, D.: Base: An acid alternative. Queue 6(3), 48–55 (2008)
22. Shoup, R.: Being Elastic - Evolving Programming for the Cloud. Presentation at QCon San Francisco (November 2010)
23. Wei, Y., Sukumar, K., Vecchiola, C., Karunamoorthy, D., Buyya, R.: Aneka cloud application platform and its integration with windows azure. CoRR, abs/1103.2590 (2011)

Part VI
Clustering and Classification

Mixed Clustering Methods to Forecast Baseball Trends

Héctor D. Menéndez*, Miguel Vázquez, and David Camacho

Abstract. Sport betting has become one of the most profitable business around the world. This business generates millions of dollars every year. One of the most influenced games is Baseball. Baseball has suffered an important change after the introduction of statistical methods to tune up the team strategy. This effect, called Moneyball, started in 2002 when the team Oaklans Atletics began to choose players according to their statistics. After this successful approach, several teams decided to continue with this strategy, generating strong statistical teams. The statistical information about players and matches have acquired highly importance, creating different datasets, such as Retrosheet which collects detailed information about players, teams and matches since 1956 until today. This work pretends to generate a forecasting model for Baseball focused on the result prediction of new matches using statistical previous information. We combine time-series and clustering algorithms to generate a model which learns about the teams and matches evolution and tries to predict the final results. Even whether this model is not complete accurated, it becomes a good starting point for future models.

Keywords: Clustering, Time Series, Forecast, Baseball.

Héctor D. Menéndez · Miguel Vázquez · David Camacho
Departamento de Ingeniería Informática, Escuela Politécnica Superior,
Universidad Autónoma de Madrid,
C/Francisco Tomás y Valiente 11, 28049 Madrid, Spain
e-mail: {hector.menendez,david.camacho}@uam.es,
miguel.vazquezf@estudiante.uam.es
http://aida.ii.uam.es

* This work has been partly supported by: Spanish Ministry of Science and Education under project TIN2010-19872 and Savier an Airbus Defense & Space project (FUAM-076914 y FUAM-076915). The information used here was obtained from Retrosheet (copyright). Interested parties may contact Retrosheet at 20 Sunset Rd., Newark, DE 19711.

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_19

1 Introduction

Sport Forecast is one of the most challenging problems. The problem consist on predict match results based on a dataset extracted from different teams, players and matches of a concrete sport. One of the most difficult steps for this process is to find the most appropriate dataset. Usually, some sport datasets contains general season information while other contains more detailed information about the play-by-play, game logs, players aligment, etc. Baseball is one of the sports which contains more detailed data about the different teams and players. There exists a dataset called Retrosheet[1] which contains lots of information about player, teams and matches. They accumulate the information using game logs of the different events during a match.

Several techniques such as statistics, data mining and machine learning have been used to analyse the performance of teams and players in games like soccer [8], football, basket, etc. These approaches, usually named human or robot behaviour modelling, has been applied in different domains like Robosoccer simulations [5], but, in these examples, all the information is totally controlled and simulated. Other similar analysis applied to human team games can be found in the NBA league. Vaz de Melo et al. [7] analyse the evolution of this league during its whole history creating a complex network model and studying its evolution. To the football or soccer analysis problem, Onody and Castro [9] propose a model also based on complex networks but only applied to analyse Brazilian players and Bitter et al. [1] generates a statistical model, modifying classical probability distribution such as Bernouilli and Gaussian distribution to create a score model for different leagues.

From the baseball point of view, there are also several works which deal with the forecasting problem. In [2] they propose a visualization framework for baseball to extract information about different teams and matches in order to query different aspects of baseball. Hakes and Sauer [4] study the Moneyballl effect from an economic perspective. Their goal was to prove that there was an inefficiency players evaluation for baseball market over a prolonged period of time. Exploitation of this inefficiency by the Oakland Athletics team suppose an outstanding progress for the baseball strategies. Other research is focused on different analytical perspectives, for example, Marchi and Albert [6] introduces several techniques to analyse the different parts of a baseball macth, team, player, etc, using R. They provide several analytical methods extracted from mathematics. They also introduce some machine learning methodology but only focused on classification and regression.

This work have been focused on a new perspective to produce a model for baseball forecasting. This model studies the evolution of the teams and matches using time series and clustering to provide a forecasting model. The model tries to generate two graph: a team similarity graph and a match similarity graph, which are related to each other. Using the information of these graph, the model tries to predict the results of a new match using information of the teams which are playing and their previous matches. The model has been train using data about 2003 and 2004 baseball seasons and it has been tested using data of 2005 season.

[1] http://retrosheet.org/

The rest of the paper is structured as follows: Section 2 defines the prediction methodology; Section 3 shows the experiments which have been carried out, and, finally, the last section provides the conclusions and future work.

2 Prediction Methodology

The prediction methodology can be divided in the two main steps: Model Generation (the data to train the model are chosen, and it is trained), and Match Prediction (the model is applied to a new match).

2.1 Model Generation Phase

The model generation phase can be summarized in four main steps:

1. The time series statistics of the teams and matches are generated. For teams, the temporal statistics are generated by match, in this case, each chosen variable to be considered in the search space can be represented (see Fig. 1), this allows to compare the teams evolution during a season. For matches, the temporal statistics are generated by inning, this helps to compare matches evolution from its beginning to end.

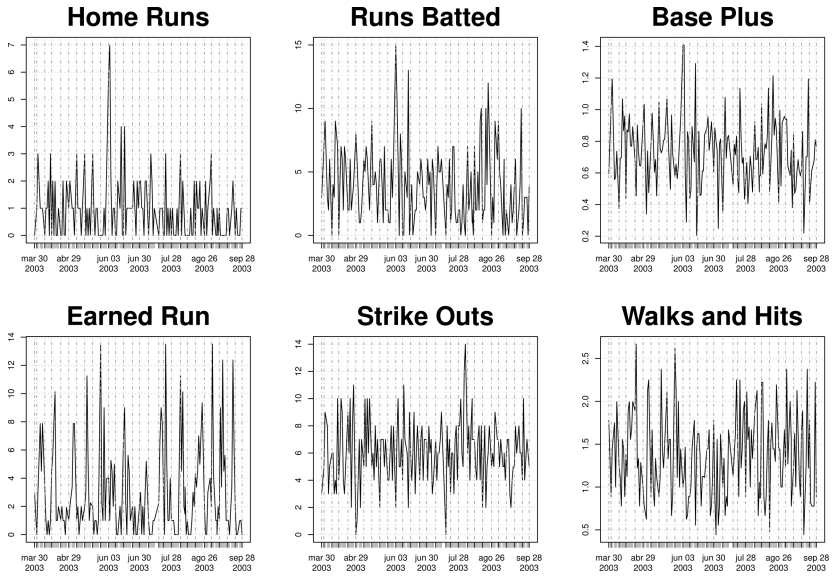


Fig. 1 Time series of the 6 chosen metrics for team "Los Angeles Angels of Anaheim" (ANA) during 2003

2. We have clusterized the time series of the teams per metric, using this information, we generate a matrix with the metric cluster and the team associated (see Fig. 2). The same procedure is followed for matches.

Metric	Team 1	Team 2	Team 3	...	Team N
Home Runs	C_1^{hr}	C_1^{hr}	C_2^{hr}	...	C_5^{hr}
Runs Batted	C_1^{rb}	C_2^{rb}	C_2^{rb}	...	C_1^{rb}
Base Plus	C_1^{bp}	C_3^{bp}	C_2^{bp}	...	C_5^{bp}
Earned Run	C_1^{er}	C_1^{er}	C_3^{er}	...	C_2^{er}
Strike Outs	C_1^{so}	C_4^{so}	C_2^{so}	...	C_2^{so}
Walks and Hits	C_1^{wh}	C_1^{wh}	C_4^{wh}	...	C_3^{wh}

Fig. 2 Example of the clusters matrix of the 6 chosen metrics for all teams. The labels are usually integer number corresponding to the clustering assignation.

3. Using the previous matrix, a similarity graph around the teams is generated, and this similarity graph will be used to compare the teams among them (see Fig. 3, Teams). The similarity measure applied for teams is the following:

$$sim(t_i, t_j) = \frac{\sum_{C_q} \delta^i_{C_q} \cdot \delta^j_{C_q}}{M} \tag{1}$$

Where M is the number of metrics considered, t_i, t_j are the teams to be compared, C_q represents the possible clusters per metric, and $\delta^i_{C_q}$ defines the Dirichlet delta defined by:

$$\delta^i_{C_q} = \begin{cases} 1 & \text{if } t_i \in C_q \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

4. Matches has a similar process than teams, however, the similarity graph includes the information about the teams in its metric (see Fig. 3, Teams). The similarity measure between two matches is calculated as follows:

$$sim(m_i, m_j) = \frac{1}{2} \cdot \frac{\sum_{C_q} \delta^i_{C_q} \cdot \delta^j_{C_q}}{M} + \frac{1}{2} \max_{i,j} \left(\frac{\sum_{k=1}^{2} sim(t_i^k, t_j^k)}{2} \right) \tag{3}$$

where m_i, m_j represents the matches, and the second factor of the sum, represents the maximum similarity between the two teams of the two matches according to the teams similarity metric. It is important to keep the teams information in the model when the matches are compared, because the match information is unknown in the prediction phase, the only information that is known is the previous team information.

2.2 *Match Prediction Phase*

The match prediction phase can be summarized in the following steps:

1. The two teams which are going to play the match are compared with the teams of the model. To compare the different teams, we use first considered the last N matches of the teams which provide the most recent evolutionary statistics of the team.
2. The T most similar teams are chosen for each team. Let $\mathfrak{T}_1$ be the T most similar teams to t_1 and $\mathfrak{T}_2$ be the T most similar teams to t_2. Them, the M matches played by teams of $\mathfrak{T}_1$ against teams of $\mathfrak{T}_2$ are extracted.
3. The teams are configured in a matrix of matches and victories according to similarity degree (see Figure 4)
   The similarity degree (sdeg) is measured by:

$$sdeg([t_i, t_j], [\mathfrak{T}_i^k, \mathfrak{T}_j^q]) = \frac{1}{2} sim(t_i, \mathfrak{T}_i^k) + \frac{1}{2} sim(t_j, \mathfrak{T}_j^q) \tag{4}$$

   where $\mathfrak{T}_i^k$ is a team from set $\mathfrak{T}^i$ and $\mathfrak{T}_j^q$ is a team from set $\mathfrak{T}_j$.
4. Finally, the victory probability is calculated as:

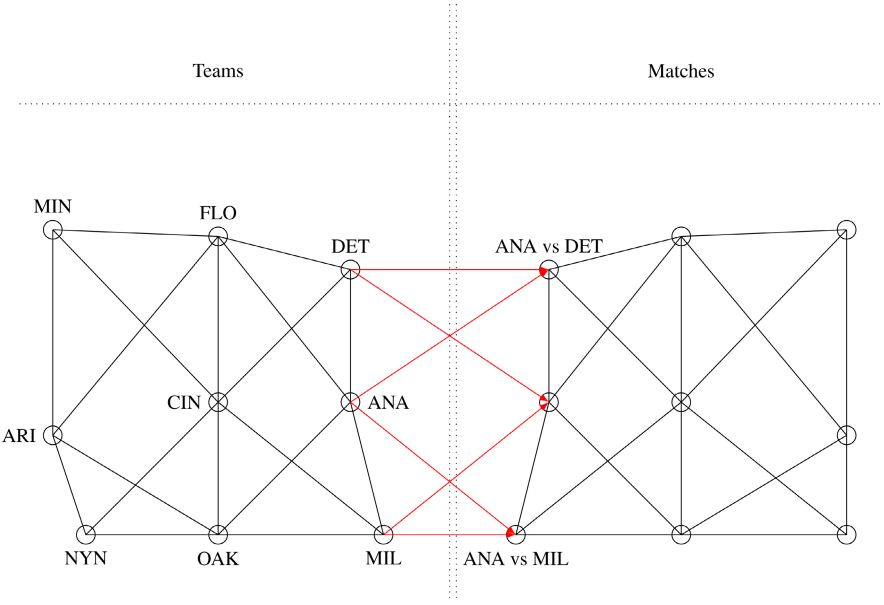


Fig. 3 Representation of the similarity graphs amongst teams and matches and the connections between these two graphs related to those teams which have played a match

Matches	Victory	Sim. Degree
$\mathcal{T}_1^1$ vs $\mathcal{T}_2^1$	$\mathcal{T}_1^1$	0.5
$\mathcal{T}_1^1$ vs $\mathcal{T}_2^2$	$\mathcal{T}_2^2$	0.8
$\mathcal{T}_1^1$ vs $\mathcal{T}_2^3$	$\mathcal{T}_1^1$	0.2
...	...	...
$\mathcal{T}_1^2$ vs $\mathcal{T}_2^1$	$\mathcal{T}_2^1$	0.3
$\mathcal{T}_1^2$ vs $\mathcal{T}_2^2$	$\mathcal{T}_1^2$	0.4
...	...	...
$\mathcal{T}_1^M$ vs $\mathcal{T}_2^{M-1}$	$\mathcal{T}_2^{M-1}$	0.8
$\mathcal{T}_1^M$ vs $\mathcal{T}_2^M$	$\mathcal{T}_1^M$	0.9

Fig. 4 Example of similarity of matches and teams including the victorius team and its similarity degree

$$V(m) = \max\left\{\frac{\sum_{q=1}^{M} \tau_{\mathcal{T}_1^q} \cdot sdeg(m)}{M}, \frac{\sum_{q=1}^{M} \tau_{\mathcal{T}_2^q} \cdot sdeg(m)}{M}\right\} \tag{5}$$

where m is the match which is compared, t_1 and t_2 τ is similar to Dirichlet delta. It is defined by:

$$\tau_{\mathcal{T}_i^q} = \begin{cases} 1 \text{ if } \mathcal{T}_i^q \text{ won} \\ 0 \text{ otherwise} \end{cases} \tag{6}$$

and

$$sdeg(m) = sdeg([t_1, t_2], [\mathcal{T}_1^k, \mathcal{T}_2^q]) \tag{7}$$

3 Experiments

This section explains the experimental results of the model generated. First, the dataset applied to the model is explain. Later, the experimental setup is presented, explaining the different variables which has been considered during the experiments and the parameters for the algorithms and model. Finally, a discussion about the results is explained to give more details about the model application.

3.1 Dataset

The Retrosheet database provides several statistics about baseball. This database contains play-by-play and game logs information about different baseball matches and leagues. They have information from 1871 until today. According to the Moneyball effect, we have decided to focused the analysis in the three years which continues the Oakland Athletics successful in 2002 season. We have chosen 2003 and 2004 as training years to train the model, and 2005 as the evaluation dataset.

3.2 Experimental Setup

Retrosheet contains several variables aboout players, teams, games and innings. In this work we have based the analysis on a subset of those variables which are more relevant from the Official Site of Major League Baseball. We have chosen the following statistics which are defined by the Official Site of Major League Baseball[2] as follows (see Fig. 2):

- (B) Home Runs: The number of times a batter hits the ball and reaches home plate scoring a run, either by hitting the ball out of play in fair territory, or without aid of an error or fielder's choice.
- (B) Runs Batted In: The number of runs that score safely due to a batter hitting a ball or drawing a base on balls.
- (B) On Base Plus Slugging: The number of times each batter reaches base by hit, walk or hit by pitch, divided by plate appearances including at-bats, walks, hit by pitch and sacrifice flies.
- (P) Earned Run Average: The average number of earned runs allowed by a pitcher; total number of earned runs allowed multiplied by 9 divided by the number of innings pitched.
- (P) Strike Outs: The number of strikeouts by a pitcher.
- (P) Walks and Hits per inning: The average number of walks and hits by a pitcher, Hits plus walks allowed divided by innings pitched.

Therefore, there are 3 batting measures (B) and 3 pittching measures (P). The parameters chosen for the model are shown in Table 1. The models have been generated using teams of seasons 2003 and 2004 (30 teams per season and a total of 4930 matches). Every team has played, at least, 161 matches. Each model has been applied 6 times to predict 100 random matches of 2005 season.

Table 1 Parameter selection for the model

Parameter	value or value range	sequences (ranges)
N (past matches)	161	-
T (sim. teams)	[3-9]	2
M (sim. matches)	[3-9], all	2

The Time Series clustering process has been carried out using a hierarchical cluster analysis on a set of dissimilarities [3]. First, the teams or matches Time Series dissimilarities have been calculated using the Correlation-based dissimilarity metric (which computes the estimated Pearson correlation of two given time series), defined by:

$$d = \sqrt{\left(\frac{1-\rho}{1+\rho}\right)^{\beta}} \tag{8}$$

[2] `http://mlb.mlb.com`

where ρ is the Pearson correlation and the parameter β has been set to 2, in order to obtain a positive value from the metric.

3.3 Results and Discussion

Table 2 shows the results of the model application. This table is divided in three parts: the first part shows the average results of the different models generated which applies 3 to 9 teams. The second part shows the average results of the different models generated using 3 to 9 matches or all matches. Finally, the last part shows all the models which has been generated in the experimental phase. Also the maximum, minimum, mean and standard deviation values of each model have been represented in this table.

Table 2 Results table divided in three sections: first section (3-N to 9-N) is a summary of the different similar teams chosen for all the matches posibilities, second section (N-3 to N-all) shows different matches chosen for all the similar teams possibilities, and finally, third section (3-3 to 9-all) shows all the models which has been designed

Teams-Matches	Max	Min	Mean	SD
3-N	59.76%	***41.76*** %	50.00%	± 0.0529
5-N	***62.24%***	40.00%	51.09%	± 0.0517
7-N	60.44%	**42.22%** Âž	***52.27%***	± 0.0460
9-N	**63.44%**	38.30%	**53.68%**	± 0.0611
N-3	**63.44%**	**42.53**%	48.95%	± 0.0571
N-5	***62.77%***	40.00%	50.83%	± 0.0525
N-7	62.64%	38.30%	**53.76%**	± 0.0545
N-9	60.44%	***42.22%***	***51.87%***	± 0.0499
N-all	59.76%	39.78%	50.86%	± 0.0521
3-3	57.95%	42.53%	46.51%	± 0.0551
3-5	52.81%	47.19%	48.84%	± 0.0200
3-7	59.14%	41.76%	54.12%	± 0.0639
3-9	55.91%	42.70%	54.35%	± 0.0575
3-all	59.76%	42.17%	52.17%	± 0.0661
5-3	62.24%	43.16%	52.75%	± 0.0661
5-5	54.26%	40.00%	46.81%	± 0.0492
5-7	54.95%	47.87%	52.17%	± 0.0273
5-9	60.44%	47.87%	51.11%	± 0.0399
5-all	57.14%	42.22%	50.54%	± 0.0589
7-3	56.84%	43.33%	48.45%	± 0.0462
7-5	59.18%	43.96%	52.27%	± 0.0506
7-7	60.00%	***51.06%***	53.41%	± 0.0364
7-9	60.44%	42.22%	50.52%	± 0.0575
7-all	54.74%	46.32%	51.16%	± 0.0272
9-3	**63.44%**	45.45%	50.54%	± 0.0665
9-5	***62.77%***	**51.58%**	**56.12%**	± 0.0433
9-7	62.64%	38.30%	***56.04%***	± 0.0785
9-9	55.43%	42.55%	53.76%	± 0.0524
9-all	56.52%	39.78%	50.55%	± 0.0568

The average results for each model applied to different teams values show those models which use 7 and 9 similar teams obatin better mean results (52.27% and 53.68% respectively). According to the best models, they can be found in 5 and 9 teams model (the maximum values are 62.24% and 63.44%, respectivily). According to the standard deviation, the most stable model is the 7 similar teams model (its standard deviation is 0.0460).

The matches variation shows that the best mean models are 7 and 9 similar matches models (53.76% and 51.87%, respectivily). However, the models with the highest maximum values are 3 and 5 similar matches model (63.44% and 62.77%, respectivily). The "all similar matches" model does not achieve remarkable results in any case, which suggest that it is not relevant to include all the information, but it is enough to use relevant and more reduced information sources. The most stable model is the 9 similar matches model (0.0499 of standard deviation).

The general models show more details about the quality of the parameters selection. The worse mean models are those based on a small number of teams (3 and 5, specially) and a small number of similar matches (also 3 and 5). This is really representative for models 3-3 and 3-5. The best model usually uses a balanced between these values, in this case 9-5 and 9-7 obtain the best average results (56.12% and 56.04%) however, the most stable of these two models is 9-5 (with a standard deviation of 0.0433). The best model, according to the maximum, are 9-3 and 9-5 (63.44% and 62.77%). According to the minimum value, 9-5 and 7-7 are the models with highest minimum. The best model of the analysis is clearly 9-5 because it obtains good results for all the parameters and it is also stable.

These results have shown that the models requires a balanced information about the number of similar teams and number of similar matches which is considered in the model generation phase. If all the information is included, the model does not obtain good results and, when there is lack of information, the model is not able to predict with high accuracy results.

4 Conclusions and Future Work

This paper presents a forecasting model based on different approaches extracted from clustering. The model combines time-series clustering and graph theory to generate a matches and teams model in order to use it to predict the results of a match. The model has been compared using different parameters in order to achieved the best parameter selection. These different models has shown that the best configuration needs a balanced selection between the number of similar teams and similar matches which are considered during the prediction phase.

The future work will be focused on different improvements of the model: first, different time series metric can be evaluated in order to achieved better results; second, other parameters such as player alignation, attitude, motivation, injuries, etc. can be considered in the model development; third, more metrics (apart the Pearson correlation) can be considered for the model generation; and, finally, a confidence factor can be included in order to provide information about the reliability of the prediction.

References

1. Bittner, E., NuBbaumer, A., Janke, W., Weigel, M.: Self-affirmation model for football goal distributions. EPL (Europhysics Letters) 78(5), 58002 (2007), http://stacks.iop.org/0295-5075/78/i=5/a=58002
2. Cox, A., Stasko, J.: Sportsvis: Discovering meaning in sports statistics through information visualization. In: Compendium of Symposium on Information Visualization, pp. 114–115. Citeseer (2006)
3. Everitt, B.: Cluster analysis. Reviews of current research. Heinemann Educational [for] the Social Science Research Council (1974), http://books.google.es/books?id=KjQNAQAAIAAJ
4. Hakes, J.K., Sauer, R.D.: An economic evaluation of the moneyball hypothesis. The Journal of Economic Perspectives 20(3), 173–185 (2006)
5. Jiménez-Díaz, G., Menéndez, H.D., Camacho, D., González-Calero, P.A.: Predicting performance in team games. In: INSTICC - Institude for systems and Technologies of Information, Control and Communication (ed.) Proceedings of the 3rd International Conference on Agents and Artificial Intelligence, ICAART 2011, vol. 1, pp. 401–406 (2011), http://aida.ii.uam.es/wp-content/uploads/2011/06/icaart_2011.pdf
6. Marchi, M., Albert, J.: Analyzing Baseball Data with R. CRC Press, Taylor and Francis Group (2013)
7. Vaz de Melo, P.O., Almeida, V.A., Loureiro, A.A.: Can complex network metrics predict the behavior of nba teams? In: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2008, pp. 695–703. ACM, New York (2008), doi:http://doi.acm.org/10.1145/1401890.1401974
8. Menendez, H., Bello-Orgaz, G., Camacho, D.: Extracting behavioural models from 2010 fifa world cup. Journal of Systems Science and Complexity 26(1), 43–61 (2013), http://link.springer.com/article/10.1007%2Fs11424-013-2289-9
9. Onody, R.N., de Castro, P.A.: Complex network study of brazilian soccer players. Phys. Rev. E 70, 037103 (2004), http://link.aps.org/doi/10.1103/PhysRevE.70.037103, doi:10.1103/PhysRevE.70.037103

SACOC: A Spectral-Based ACO Clustering Algorithm

Héctor D. Menéndez*, Fernando E.B. Otero, and David Camacho

Abstract. The application of ACO-based algorithms in data mining is growing over the last few years and several supervised and unsupervised learning algorithms have been developed using this bio-inspired approach. Most recent works concerning unsupervised learning have been focused on clustering, where ACO-based techniques have showed a great potential. At the same time, new clustering techniques that seek the continuity of data, specially focused on spectral-based approaches in opposition to classical centroid-based approaches, have attracted an increasing research interest—an area still under study by ACO clustering techniques. This work presents a hybrid spectral-based ACO clustering algorithm inspired by the ACO Clustering (ACOC) algorithm. The proposed approach combines ACOC with the spectral Laplacian to generate a new search space for the algorithm in order to obtain more promising solutions. The new algorithm, called SACOC, has been compared against well-known algorithms (K-means and Spectral Clustering) and with ACOC. The experiments measure the accuracy of the algorithm for both synthetic datasets and real-world datasets extracted from the UCI Machine Learning Repository.

Keywords: Clustering, Data Mining, ACO, Spectral.

1 Introduction

Unsupervised data mining techniques compose a complex field, where several different approaches have been tested in order to obtain similar or even better results to supervised techniques. The main difference between these two techniques is that

Héctor D. Menéndez · David Camacho
Departamento de Ingeniería Informática, Universidad Autónoma de Madrid, Spain
e-mail: {hector.menendez,david.camacho}@uam.es

Fernando E.B. Otero
School of Computing, University of Kent, UK
e-mail: F.E.B.Otero@kent.ac.uk

* This work has been partly supported by: Spanish Ministry of Science and Education under project TIN2010-19872 and Savier an Airbus Defense & Space project (FUAM-076914 and FUAM-076915).

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_20

supervised techniques have the label (target) information, which is used during the model generation, providing a more accurate model—the accuracy of the model is determined by comparing the prediction with the label information. Unsupervised techniques, instead, are totally blind in respect to the label information. An advantage of unsupervised techniques is that they can deal with a huge quantity of (unlabeled) data without a feedback of their performance.

Unsupervised techniques have been studied from different perspectives. Over the last few years, bio-inspired techniques are the most representatives, usually based on evolutionary algorithms or swarm intelligence that mimic a natural behaviour—e.g., the evolutionary process in genetic algorithm, collective behaviour in ant colony optimization. This work has been focused on the latter, which is becoming a promising field for unsupervised techniques. ACO algorithms are based on the foraging behaviour of ant colonies when they try to find the optimal path between their nest and a food source. Based on this idea, researchers have created several optimization algorithms in data mining, which have been focused on the path optimization process followed by the ants to create solutions for hard optimization problems [10, 14, 15].

The work presented in this paper is focused on the application of ACO in the unsupervised learning task of clustering, where the goal is to group (cluster) similar data points in the same group and, at the same time, maximise the difference between different clusters. It has been inspired by the Spectral Clustering (SC) algorithm [12] and the ACO-based Clustering algorithm (ACOC), proposed by Kao and Cheng [6]. ACOC is a centroid-based clustering algorithm, which tries to optimize the centroid (central point) position of each cluster. Following this idea, we focused the proposed algorithm on addressing a spectral-based approach. Inspired by other clustering algorithms [11, 13], we reformulated the original ACOC algorithm to create a spectral-based algorithm. Spectral-based clustering algorithms are usually good to define continuity-based clusters. They usually work with similarity graph amongst the data instances, which can be obtained as a Gram matrix of a kernel or a distance measure, and they study the spectrum of the graph in order to find the best cluster discrimination. In order to check the performance of the proposed algorithm, we have compared it against well-known clustering algorithms SC (Spectral Clustering) and [12] and K-means [9], as well as the original ACOC algorithm, in synthetic and real-world datasets.

The rest of the paper is structured as follows: Section 2 introduces the related work, Section 3 presents the new algorithm, Section 4 presents the computational results on synthetic and real-world datasets, and, finally, the last section discusses the conclusions and future work.

2 Related Work

Ant Colony Optimization (ACO) has become a promising field for data mining problems. In this context, ACO algorithms combine the ants foraging behaviour to generate patterns that describe the data according to a supervised or unsupervised learning

criteria—depending on the type of algorithm, classification or clustering, respectively. This paper focuses on clustering problems.

Clustering [7] is based on a blind search within the data. Clustering techniques try to join similar data points into groups (clusters) according to a cost or objective function, which is usually minimized or maximized, making this clusters different from each other at the same time. There is a large number of clustering approaches depending on the goal that the algorithm should achieve. The most classical algorithms are K-means [9] and EM [3]. Both K-means and EM usually try to optimize estimator parameters to define clusters. Over the last decades, new non-parametrical algorithms such as Spectral Clustering [8] are gaining prominence. These algorithms study the graph spectrum generated by a similarity graph, usually extracted from a Kernel function and the Gram matrix associated by the application of the kernel to the data instances. The study of the spectrum maps the original data points to a projective space, where a simple K-means can be applied to group the data into clusters.

There are also bio-inspired algorithms that deal with the clustering problem, several of them focused on genetic algorithms. Hruschka et al. [4] presents a survey of clustering algorithms from different genetic approaches. From other bio-inspired perspectives, ACO algorithms have also produced promising results. Kao and Cheng [6] introduced a centroid-based ACO clustering algorithm; and Ashok and Messinger focused their work on graph-based clustering [1]; several other approaches are discussed in [5].

3 Spectral-Based ACO Clustering Algorithm (SACOC)

This section presents the proposed Spectral-based ACO Clustering Algorithm (SACOC). This algorithm is similar to Spectral Clustering. The goal of the algorithm is to choose the data discrimination representing the information as a similarity graph and cutting it in different clusters.

3.1 ACOC Algorithm

The ACOC algorithm is the base of SACOC. It has a search space based on instances and centroids, and can be defined as a graph whose associated matrix is a N×M matrix, where N is the number of instances and M is the number of centroids (clusters).

The algorithm works with several ants looking for the best path in the graph Each ant (k) has the following features: a list of visited objects (tb^k), a set of chosen centroids C^k and a Weighted matrix W^k (related to the assignation of objects to clusters).

An ant k has two possible strategies: exploration and exploitation. It choose the strategy according to the following formula:

$$j = \begin{cases} argmax_{u \in N_i}\{[\tau(i,u)][\eta^k(i,u)]^\beta\} & , \text{ if } q \leq q_0 \\ S & , \text{ otherwise} \end{cases} , \quad (1)$$

where N_i is the set of nodes associated to object i, j is the chosen cluster, $\tau(i,u)$ is the pheromone value between i and u, q_0 is the exploitation probability, q is a random number for strategy selection,β is a parameter, $\eta^k(i,u)$ is the heuristic value between i and u for ant k defined by the formula:

$$\eta^k(i,u) = 1/d(x_i, c_j^k) = ||x_i - c_j^k|| \ , \quad (2)$$

where x_i is a data instance and c_j^k is a centroid from the ant centroid list. and S is the exploration defined by:

$$S = P^k(i,u) = \frac{[\tau(i,u)][\eta^k(i,u)]^\beta}{\sum_{j=1}^{m}[\tau(i,j)][\eta^k(i,j)]^\beta} \ . \quad (3)$$

The algorithm steps can be divided by:

1. Initialize pheromone matrix.
2. Initialize ants: (tb^k, C^k, W^k), for each ant k in the colony. Then, each ant repeats until tb^k is full:
   a. Select (randomly) a data object i satisfying $i \notin tb^k$.
   b. Select a cluster j: first the ant chooses a strategy; then, it calculates the transition probability and, finally, it visits a node.
   c. Update tb^k, C^k and W^k.
3. Choose the best solution. First, calculate the objective function for each ant:

$$J^k = \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}^k d(x_i, c_j^k) \ , \quad (4)$$

   where w_{ij}^k is a weight value of the assignation matrix W^k. Next, rank ants solutions. Choose the iteration-best solution, apply local search[1] to improve the solution and, finally, compare it with the best-so-far solution and update this value with the maximum between them.
4. Update pheromone trails (global updating rule). Only the best r ants are able to add pheromones. Let ρ be the pheromone evaporation rate, ($0 < \rho < 1$), t the iteration number, r is the number of elitism ants and $\Delta\tau_{ij}^h = 1/J^h$:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{h=1}^{r} w_{ij}^h \Delta\tau_{ij}^h \ . \quad (5)$$

5. Check termination condition: if the number of iterations is greater than the maximum limit, finish; otherwise, go to step 2.

[1] For more details of local search see [16].

3.2 The Spectral Hybridisation

The original ACOC algorithm uses the euclidean space as a search space. However, the algorithm can be modified to consider any kernel in a similar way that K-means is modified to generate the Spectral Clustering algorithm. Consider a graph G and its associated weighted matrix W, which is a pairwise similarity graph amongst the data. The similarity is calculated using a similarity function defined by a kernel $k(x_i, x_j)$. The Spectrum of the graph is calculated in a similar fashion used by Ng et al. [12] to create the original Spectral Clustering algorithm. First, we calculate the Laplacian matrix defined by:

$$L = I - D^{-1/2} W D^{-1/2} \ , \tag{6}$$

where I is the identity matrix and D represents the diagonal matrix whose (i, i)-element is the sum of the similarity matrix i-th row. After the creation of the Laplacian matrix, we extract the $v_1, \ldots, v_z$, which corresponds with the z largest eigenvectors of L—chosen to be orthogonal to each other in the case of repeated eigenvalues—and form the matrix $V = [v_1\, v_2\, \ldots\, v_z] \in \mathbb{R}^{n \times z}$ by stacking the eigenvectors in columns. Finally, we form the matrix Y from V by renormalizing each row of V to have unit length (i.e., $Y_{ij} = V_{ij}/(\sum_j V_{ij}^2)^{1/2}$). Then, we can consider Y as a projection of the original space and apply ACOC to the representation of each point.

4 Experiments

This section shows the experimental results. First, the synthetic and real-world datasets are described. Then, the experimental setup is shown. Finally, the computational results for both synthetic and real-world datasets are discussed.

4.1 Datasets Description

For the synthetic experiments we have used the following datasets [11]:

- *Aggregation*: This dataset is composed by 7 clusters, some of them can be separated by parametric clustering;
- *Jain*: This dataset is composed by two surfaces with different density and a clear separation;
- *Spiral*: In this case, there are 3 spirals close to each other.

For the real-world experiments, we have chosen three datasets from UCI Machine Learning Repository [2]:

- *Iris*: Contains 50 instances distributed over 3 classes, with 4 attributes each;
- *Haberman*: Contains 306 instances distributed over 2 classes, with 3 attributes each;
- *Breast Tissue* (Bre. Tis.): Contains 106 instances distributes over 6 classes, with 10 attributes each.

4.2 Experimental Setup

We have chosen K-means [9], Spectral Clustering (SC) [12] and ACOC [6] clustering algorithms to compare the results of SACOC.[2] K-means is an iterative algorithm based on centroids. The goal of the algorithm is to find the best centroids position. It involved two steps: it assigns the data to the closest centroid (cluster) and then, it calculates the new position of the centroid as a centroid of the data which has been assigned to it. SC generates a similarity graph and extracts its spectrum as a projective space in order to applied a simple clustering algorithm (in this case K-means) to the projective data.

The parameters of ACOC and SACOC are: the ants number is 10, the elitism is 1, the exploitation probability is 0.0001, the initial pheromone values have been set to $1/m$—where m is the number of clusters, $\beta = 2.0$, $\rho = 0.1$, the local search probability is 0.001 and the maximum number of iterations is 1000. These values have been chosen according to the original ACOC paper [6].

All algorithms need the number of cluster as an initial parameter. The experiments have been carried out 50 times using the Euclidean distance as the metric, except for Spectral Clustering and SACOC which use the Radial Basis Function. The evaluation of the experiments has been focused on two different ideas: the synthetic datasets have been evaluated according to the cluster discrimination and the performance of the algorithm in discriminating the original clusters; the real-world datasets have been evaluated using the accuracy rate, in order to check how close the algorithm is to real criteria.

4.3 Synthetic Experiments

Figure 1 presents the visual (best) results of the SC and SACOC algorithms when applied to the synthetic datasets. Table 1 shows the accuracy results on the same datasets.

[2] We used the K-means and SC implementation available in R; the author's implementation of ACOC and SACOC is available upon request.

Table 1 Minimum, Maximum, Median, Mean and Standard Deviation accuracy results of the application of the algorithms to the synthetic datasets. The p-values for the Wilcoxon test applied to SACOC and SC results are: Aggregation ($p = 1.062 \times 10^{-8}$), Jain ($p = 0$) and Spiral ($p = 0.02225$)—statistical significant improvements are indicated by a ▲ symbol

SACOC	Min	Max	Median	Mean	SD
Aggregation	**98.60**%	**99.62**%	**99.24**%	**99.28**%	▲ ± 0.0022
Jain	**100.0**%	**100.0**%	**100.0**%	**100.0**%	± 0.0000
Spirals	**100.0**%	**100.0**%	**100.0**%	**100.0**%	▲ ± 0.0000
SC	Min	Max	Median	Mean	SD
Aggregation	63.96%	99.37%	88.39%	90.30%	± 0.0716
Jain	**100.0**%	**100.0**%	**100.0**%	**100.0**%	± 0.0000
Spirals	35.26%	**100.0**%	**100.0**%	93.20%	± 0.1724
K-means	Min	Max	Median	Mean	SD
Aggregation	66.88%	88.07%	78.55%	77.93%	± 0.0495
Jain	78.28%	78.28%	78.28%	78.28%	± 0.0000
Spirals	33.97%	34.94%	34.29%	34.41%	± 0.0020
ACOC	Min	Max	Median	Mean	SD
Aggregation	62.18%	86.17%	77.73%	77.07%	± 0.0516
Jain	73.19%	76.68%	74.80%	74.97%	± 0.0067
Spirals	33.65%	36.54%	35.26%	35.14%	± 0.0063

Aggregation results show that SACOC achieved the best results and outperforms all the other algorithms—SACOC results are statistically significantly better than SC ($p = 1.062 \times 10^{-8}$). K-means and ACOC usually have the worse results in this dataset. These algorithms are not able to define clusters on the left (see Fig. 1), where the cluster boundaries are not clear.

Jain results show that both SC and SACOC are able to discriminate the clusters in all cases (see Table 1), both algorithms achieving the same results without statistically significant differences between them. K-means achieves stable results (the standard deviation is 0), while ACOC obtains the worst results.

Spirals shows that both SC and SACOC are able to define the clusters continuity—SACOC achieved the best and most stable results (0 standard deviation), which are statistically significantly better than SC ($p = 0.02225$). K-means and ACOC are not able to define the continuity of the data due to the use of the Euclidean space.

Overall, the results for the synthetic datasets show that SACOC achieved best results, with statistically significant differences when compared to SC. In the next section we will compare the algorithms in real-world datasets.

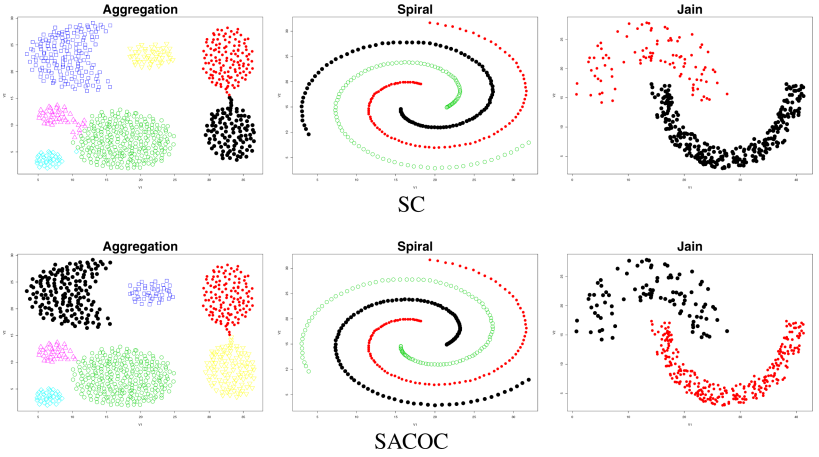


Fig. 1 Graphical representation of the best results on the synthetic datasets

Table 2 Minimum, Maximum, Median, Mean and Standard Deviation accuracy results of the application of the algorithms to the synthetic datasets. The p-values for the Wilcoxon test applied to SACOC and SC results are: Breast Tissue ($p = 7.689 \times 10^{-9}$), Haberman ($p = 3.919 \times 10^{-10}$) and Iris ($p = 5.371 \times 10^{-8}$)—statistical significant improvements are indicated by a ▲ symbol

SACOC	Min	Max	Median	Mean	SD
Breast Tissue	**39.62**%	**60.38**%	**48.11**%	**48.43**%	▲ ± 0.0432
Haberman	**73.53**%	73.53%	**73.53**%	**73.53**%	▲ ± 0.0000
Iris	66.67%	68.67%	68.67%	68.53%	▲ ± 0.0038
SC	Min	Max	Median	Mean	SD
Breast Tissue	36.79%	48.11%	41.51%	41.30%	± 0.0321
Haberman	51.31%	**75.82**%	52.12%	52.37%	± 0.0341
Iris	68.00%	68.00%	68.00%	68.00%	± 0.0000
K-means	Min	Max	Median	Mean	SD
Breast Tissue	33.02%	34.91%	33.02%	33.02%	± 0.0032
Haberman	50.00%	52.29%	51.96%	51.52%	± 0.0020
Iris	58.00%	89.33%	89.33%	84.95%	± 0.1098
ACOC	Min	Max	Median	Mean	SD
Breast Tissue	30.19%	40.57%	33.02%	33.42%	± 0.0184
Haberman	50.65%	52.94%	51.96%	51.90%	± 0.0048
Iris	**89.33**%	**92.67**%	**90.00**%	**90.23**%	± 0.0079

4.4 Real-World Experiments

Table 2 shows the results of the algorithms applied to real-world datasets from UCI Machine Learning repository [2].

Breast Tissue dataset is more a spectral-like dataset. The data is continuous and the clusters do not intersects in several parts. In this case, both SACOC and SC achieved good results—SACOC results are statistically significantly better than SC ($p = 7.689 \times 10^{-9}$). K-means and ACOC have problems in discriminating the clusters information.

In Haberman case, SACOC achieved the best results, however, SC achieves the highest maximum value. This datasets shows more stable results for SACOC than SC (the standard deviation of SACOC is 0). There is also a high statistical significance between them ($p = 3.919 \times 10^{-10}$). K-means and ACOC achieved the worse results again in this case.

Iris dataset shows intersecting results. The best results are achieved by ACOC and K-means, while SACOC and SC achieved the worse results. This problem is likely due to the data projection, since it affects both SC and SACOC. Usually, when there are places with cluster intersections, the data projection is generally—it worse produces a big cluster and a cluster with a couple of outliers. Even in this case, SACOC discriminates the clusters better than SC with statistically significant differences ($p = 5.371 \times 10^{-8}$).

These results show that SACOC achieved better and more stable results than SC in the datasets where the cluster assignation has clear boundaries and low cluster intersection. However, when there are intersection, it is harder for the algorithm to discriminate the data—in the same way that it is harder for SC.

5 Conclusions and Future Work

This paper presented a transformation of the ACOC algorithm into a Spectral algorithm. The new algorithm, called SACOC, uses spectral transformations of the original search space in order to apply the clustering in the projective space. The transformation consists on converting the original data in a graph-based representation (through a similarity graph) and calculate its Laplacian matrix. Once the Laplacian has been obtained, the eigenvectors are extracted and normalized to generate the projective space.

The proposed SACOC algorithm showed good results for synthetic datasets. It is able to discriminate continuity-based clusters with more stable results, when compared to Spectral Clustering (SC). Also, the SACOC shows good results for real datasets, except in those cases where there are cluster intersections. In this situation, it has the same problems to discriminate the data than SC.

The future work will be focused on some improvements of the algorithm, such as, an initial centroid selection and to improve the spectral projections of the algorithm, in order to avoid cluster intersection problems. Also we will study a comparison on the algorithms performance, in order to improve memory consumption and running time.

References

1. Ashok, L., Messinger, D.W.: A spectral image clustering algorithm based on ant colony optimization, pp. 83,901P–83,901P–10 (2012)
2. Bache, K., Lichman, M.: UCI machine learning repository (2013), http://archive.ics.uci.edu/ml
3. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39(1), 1–38 (1977)
4. Hruschka, E., Campello, R., Freitas, A., de Carvalho, A.: A survey of evolutionary algorithms for clustering. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 39(2), 133–155 (2009)
5. Jafar, O.M., Sivakumar, R.: Ant-based clustering algorithms: A brief survey. International Journal of Computer Theory and Engineering 2, 787–796 (2010)
6. Kao, Y., Cheng, K.: An ACO-based clustering algorithm. In: Dorigo, M., Gambardella, L.M., Birattari, M., Martinoli, A., Poli, R., Stützle, T. (eds.) ANTS 2006. LNCS, vol. 4150, pp. 340–347. Springer, Heidelberg (2006)
7. Larose, D.T.: Discovering Knowledge in Data. John Wiley & Sons (2005)
8. Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17(4), 395–416 (2007)
9. Macqueen, J.B.: Some methods of classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
10. Martens, D., Baesens, B., Fawcett, T.: Editorial survey: swarm intelligence for data mining. Machine Learning 82(1), 1–42 (2011)
11. Menéndez, H.D., Barrero, D.F., Camacho, D.: A genetic graph-based approach for partitional clustering. Int. J. Neural Syst. 24(3) (2014)
12. Ng, A., Jordan, M., Weiss, Y.: On Spectral Clustering: Analysis and an algorithm. In: Dietterich, T., Becker, S., Ghahramani, Z. (eds.) Advances in Neural Information Processing Systems, pp. 849–856. MIT Press (2001)
13. Orgaz, G.B., Menéndez, H.D., Camacho, D.: Adaptive k-means algorithm for overlapped graph clustering. Int. J. Neural Syst. 22(5) (2012)
14. Otero, F., Freitas, A., Johnson, C.: Inducing decision trees with an ant colony optimization algorithm. Applied Soft Computing 12(11), 3615–3626 (2012)
15. Otero, F., Freitas, A., Johnson, C.: A New Sequential Covering Strategy for Inducing Classification Rules With Ant Colony Algorithms. IEEE Transactions on Evolutionary Computation 17(1), 64–76 (2013)
16. Shelokar, P., Jayaraman, V.K., Kulkarni, B.D.: An ant colony approach for clustering. Analytica Chimica Acta 509(2), 187–195 (2004)

Anomalous Web Payload Detection: Evaluating the Resilience of 1-Grams Based Classifiers

Sergio Pastrana, Carmen Torrano-Gimenez, Hai Than Nguyen, and Agustín Orfila

Abstract. Anomaly payload detection looks for payloads that deviate from a predefined model of normality. Defining normality requires an intelligent approach. Machine learning algorithms have been widely applied to build classifiers that distinguish normal from anomalous activity. These algorithms construct vectors of features extracted from raw payloads of a given dataset and train the classifier with them. The success of the detection highly depends on the potential of the training dataset to properly represent network traffic. In this paper we show that an adversary knowing the distribution of the dataset and the specific feature construction method may generate attack vectors evading the classifier. Particularly, in the case the classifier uses a simple feature construction method based on 1-grams, getting real-world payloads to evade the classifier is feasible. We present experimental results regarding four well-known classification algorithms, namely, C4.5, CART, Support Vector Machines (SVM) and MultiLayer Perceptron (MLP).

1 Introduction

Intrusion Detection Systems (IDSs) look for malicious activities in network and system data. Concretely, payload-based detection looks for intrusive patterns encapsulated in application data, such as web traffic, e-mails, instant messaging, etc. Due to the high complexity and variety of application data, a common approach is the use

Sergio Pastrana · Agustin Orfila
Carlos III University of Madrid, Spain
e-mail: {spastran,adiaz}@inf.uc3m.es

Carmen Torrano-Gimenez
Institute of Physical and Information Technologies, CSIC, Spain
e-mail: carmen.torrano@iec.csic.es

Hai Than Nguyen
Telenor Research, Norway
e-mail: hai.nguyen_thanh@inria.fr

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_21

of Machine Learning (ML) algorithms [7,9] to learn a model which is used in detection time to distinguish intrusive from normal payloads. ML algorithms require data represented as feature vectors, which are obtained by performing a feature construction process on raw traffic. A common approach to extract features from payloads is to use text processing methods, like n-grams [8,10], which consider all the words of size n from the text.

ML algorithms assume that both the training and testing data have similar distributions. However, due to the presence of adversaries (which is the common scenario for IDS), the research community has lately focused on designing robust ML algorithms [3,5]. In this scenario, it must be assumed that the data used for training and testing is different because of the possible changes performed by an adversary in the payloads. Biggio et al. [3] have recently presented an approach to assess the security of pattern classifiers against these attacks. Still, there is a substantial lack of experimental work exploring the problems derived from an attacker who can modify instances at will to subvert the detection function.

In this work we analyze the resilience of IDSs that use ML as classification algorithms and 1-grams as feature construction method. Although these classifiers are effective for a particular distribution of the training dataset, they may learn patterns that are specific for this distribution. We show that an adversary can reverse engineer the detection surface and discover these specific patterns. Then, she can evade the system by properly modifying some features from the attack vector. Nevertheless, it is still required to build raw payloads from these modified vectors to obtain real-world evasions, which is suitable if the feature construction process can be inverted. For example, if the feature construction uses 1-grams, the adversary only needs to include or remove single bytes wherever she chooses into the payload. Next, we summarize the main contributions of our work:

1. We conduct experiments with modern HTTP traffic containing real world attacks.
2. We study four classification algorithms that have been widely used in the research literature for IDSs.
3. We discuss the robustness of IDSs using these algorithms and 1-grams in the presence of smart adversaries. Concretely, we present a reverse engineering and evasion attack that allows to carefully modify malicious payloads and evade the IDSs.

The remainder of this document is structured as follows. Section 2 explains the experimental setup and Section 3 describes the attacks. Then, Section 4 presents the results and finally, Section 5 provides the conclusions.

2 Experimental Setup

We use the **CSIC 2010 HTTP dataset** [1], which has been successfully used for malicious payload detection in previous works [6,8]. The traffic of the dataset contains normal and anomalous requests targeted to an e-commerce web application, with different values for those web pages that includes parameters. In total 36,000 normal

requests and 25,000 anomalous requests are included. As the traffic is generated, all the requests are labeled (either as normal or anomalous). The dataset includes modern web attacks such as SQL injection or Cross-Site Scripting (XSS).

We use the following automatic **feature construction method** based on the 1-grams method for intrusion detection: for every HTTP request p, a feature vector $x(p) = (x_1, x_2, \ldots, x_{256})$ contains the number of appearances x_i of the 1-gram i in the method, path and arguments of p. After extracting the 1-grams from the dataset, we observe that from the 256 possible features (i.e., total number of ASCII characters), only 89 (34.77%) appear one or more times in the HTTP requests. Thus, each vector is composed of these 89 features.

We have conducted our experiments with IDSs using four **classification algorithms**: C4.5 and CART decision trees, SVM, and MLP. First, each IDS is trained to classify HTTP packets using labeled data with both normal and intrusive packets, using one third of the dataset. Second, the IDS is tested with a second third of the dataset (the final third of the dataset is used to test the reverse engineering attack). Table 1 shows the effectiveness of the different IDSs studied over test data in terms of the hit rate (H), i.e., the ratio of attacks detected by the system, and the false positive rate (F), i.e., the ratio of normal payloads wrongly classified as intrusions. It can be observed that they all obtain high detection rates and acceptable false positive rates, which makes them a suitable solution to detect malicious payloads.

Table 1 Detection rate (H) and false alarm rate (F) of the classification algorithms studied

	C4.5	CART	SVM	MLP
H	0.97	0.97	0.95	0.96
F	0.04	0.07	0.06	0.12

3 Description of the Attacks

We adopt an **adversarial model** where the attacker has knowledge about the distribution of the training data used by the IDSs and the feature construction method (FC). Accordingly, the adversary can generate training samples that are similar to those used by the IDSs. Both the distribution and feature construction method may be kept secret in many scenarios. In this work we do not tackle the problem of how to get this information. Indeed, many authors have assumed that this information is known by an adversary [3] or inferable by a query-response analysis [2].

Our **reverse engineering attack** aims to approximate the classifier with models that are easy to process. Such models are later used to perform evasion attacks. The adversary knows the training distribution so she is able to generate training samples and build models from them that are good approximations, in terms of the decision surface, to the IDS classifier she wants to evade. Concretely, we use Genetic Programming (GP) to obtain an approximation of the decision surface of the actual detection model. We choose GP because it outputs tree-based expressions that are

easy to understand and can be evaluated with a recursive function, where the root and intermediate nodes are mathematical and logic functions, and the leaves are terminal features. The final output of each program or individual indicates whether the payload is considered anomalous or normal. To evaluate the GP individuals, we use the fitness function shown in Equation 1, which considers both the classification error (ratio of incorrectly classified payloads), and the C_{id}, which is a modern metric used to assess the efficacy of IDSs (see details in [4]).

$$fitness = \frac{E\_class + (1 - C_{id})}{2} \tag{1}$$

The **evasion attack** uses the GP model to look for strategies that transform a malicious payload that would be classified as anomalous by the IDSs into one that is classified as normal. The evasion search is done by analyzing the features and operators from the models to look for potential vectors that evade the classifier. Then, these vectors are mapped into real payloads to evade the system using the inverse process of the feature construction. We perform the evasion search by using a top-down tree-traversal searching algorithm over the model. The final goal is to make the root node change the output from non-zero values (meaning intrusion) to zero (meaning normal). The search over the tree models provides the adversary with a set of evasion strategies, which indicate which features should be modified in order to evade the IDSs. The adversary obtains a set of modified vectors which are given as input to the studied IDS. Each of these feature vectors that passes undetected by the IDS is considered as an evasion candidate. Then, these candidates must be mapped into real payloads by inverting the feature construction algorithm. In the concrete example of 1-grams, each feature in the vector specifies the number of 1-grams in the payload. Accordingly, the adversary only has to remove or insert 1-grams (i.e. bytes) in the payload to get the desired numbers.

4 Results

In this section we first present a sample model obtained in the reverse engineering attack, which is later used to explain how the evasion is conducted. Then, we show a malicious payload that actually evades the four IDSs studied using two different evasion strategies.

In order to control the bloat of GP individuals, we have settled a maximum depth of the trees. Concretely, we have experimented with values from 1 to 10. As an example, Figure 1 shows a GP model obtained with a maximum depth of 4. The model shows in the leaves the relevant features considered for detection, like the feature F37 (the number of 'i' characters in the payloads) or the feature F39 (number of '-' characters). We next use this individual to illustrate how the search of evasion attacks is done.

The evasion search conducted over the GP models suggests possible modifications of features to evade the classifiers. For instance, the rule in Table 2 has been obtained from a search over the model shown in Figure 1. In order to evade the GP

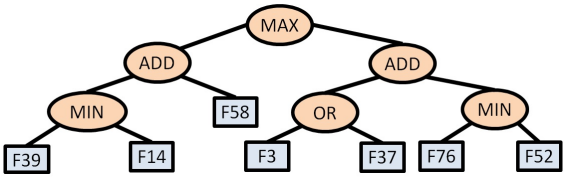


Fig. 1 Reverse Engineering attack. GP model obtained.

model, the rule expression must be evaluated to one. Thus, the rule provides different evasion strategies. For example, setting to zero the features F39 (number of '-'), F58 (number of 'A'), F37 (number of 'i'), and F76 (number of 'U'), results in vectors that evade the model. The next step is to obtain real payloads from the vectors obtained to evade the IDSs. As discussed below, this is simple because inverting the feature construction process is straightforward for the adversary.

Table 2 Sample evasion rule suggested after performing the evasion search

[(F39=0 OR F14=0) AND F58=0] AND [(F3=0 OR F37=0) AND (F76=0 OR F52=0)]

We analyze all the models generated with different tree depths, and obtain all the rule expressions. From the analysis of these rules, we next provide two examples of how an evasion could succeed. Figure 2 shows an example of malicious payload (concretely, performing an SQL Injection Attack) which was originally detected by the detectors, and after modification, evades them all. The suggested modifications, highlighted in colors in the modified payload, are:

1. **Setting the feature F37 (number of 'i') to zero.** We replace the character 'i' by its upper-case representation ('I') to generate payloads free of 'i' characters (highlighted in yellow in Figure 2). It can be observed that it is a real-world evasion because the HTTP protocol is not case-sensitive and thus the attack still succeeds. However, from the ML perspective, the corresponding feature vector of the testing data changes and so does the detection output.
2. **Setting the feature F38 (number of '-') to zero.** This requires removing the hyphens ('-') characters from the payload, which can be performed by changing these characters by underscores ('_'). In the example highlighted in green in Figure 2, the argument "email" had the value "*jperez@fighting-machines.log*". If the domain of this email is changed to "*fighting_machines.log*", the evasion suc-

```
mode=register&login=mya&password=rencor&name=Juan&surname=Perez&email=jperez@fig
hting-machines.ukAND 1=1&id=05736398Z&adress=Victoria Kents 46&town=San Miguel
de Serrezuela&cp=18330&province=Girona&ntc=2610269003230246&B1=Register');
```

```
mode=regIster&logIn=mya&password=rencor&name=Juan&surname=Perez&emaIl=jperez@fIg
htIng_machInes.ukAND 1=1&Id=05736398Z&adress=VIctorIa Kents 46&town=San MIguel
de Serrezuela&cp=18330&provInce=GIrona&ntc=2610269003230246&B1=RegIster');
```

Fig. 2 Evasion Attack. Original payload (above) and modified payload (below) which are classified as normal and intrusive respectively by the IDSs.

ceeds. However, in real settings, the HTTP request has a different semantic, i.e., the domain of the email may not exist, and the response to this request may lead to some error message, like “invalid email". Thus, in our approach, it is required to have a knowledge of which the valid modifications are in order to stealthy bypass the detection.

5 Conclusions

In this work, we present reverse engineering and evasion attacks against anomaly-based IDSs based on 1-grams and conventional machine learning classifiers. The reverse engineering attack allows discovering the patterns the detectors have learned from the training dataset. Particularly, it uses Genetic Programming to derive tree-based models. Then, the proposed evasion attack analyses these models and suggests modifications of the payload that evade the classifiers. In this case, the modifications are possible as features are easily manipulable by an adversary to get real-world evasions, due to the use of 1-grams. The premises for the attack are realistic, as both the traffic distribution of the protected network and the feature construction method are generally either public or easy to infer.

References

1. The HTTP dataset CSIC 2010 (2010)
2. Ateniese, G., Felici, G., Mancini, L.V., Spognardi, A., Villani, A., Vitali, D.: Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. arXiv:1306.4447 (2013)
3. Biggio, B., Fumera, G., Roli, F.: Security evaluation of pattern classifiers under attack. IEEE Transactions on Knowledge and Data Engineering 99(PrePrints), 1 (2013)
4. Gu, G., Fogla, P., Dagon, D., Lee, W., Skorić, B.: Measuring Intrusion Detection Capability: an Information-theoretic Approach. In: ACM Symposium on Information, Computer and Communications Security, pp. 90–101. ACM, New York (2006)
5. Huang, L., Joseph, A.D., Nelson, B., Rubinstein, B.I., Tygar, J.D.: Adversarial machine learning. In: Workshop on Security and Artificial Intelligence, pp. 43–58. ACM, NY (2011)
6. Nguyen, H.T., Torrano-Gimenez, C., Alvarez, G., Franke, K., Petrović, S.: Enhancing the effectiveness of web application firewalls by generic feature selection. Logic Journal of IGPL 21(4), 560–570 (2013)
7. Pastrana, S., Mitrokotsa, A., Orfila, A., Peris-Lopez, P.: Evaluation of classification algorithms for intrusion detection in MANETs. Knowledge-Based Systems 36, 217–225 (2012)
8. Torrano-Gimenez, C., Nguyen, H.T., Alvarez, G., Franke, K.: Combining expert knowledge with automatic feature extraction for reliable web attack detection. Security and Communication Networks (2012)
9. Tsai, C.-F., Hsu, Y.-F., Lin, C.-Y., Lin, W.-Y.: Intrusion detection by machine learning: A review. Expert Systems with Applications 36(10), 11994–12000 (2009)
10. Wang, K., Parekh, J.J., Stolfo, S.J.: Anagram: A content anomaly detector resistant to mimicry attack. In: Zamboni, D., Kruegel, C. (eds.) RAID 2006. LNCS, vol. 4219, pp. 226–248. Springer, Heidelberg (2006)

Online Gamers Classification Using K-means

Fernando Palero, Cristian Ramirez-Atencia, and David Camacho

Abstract. In order to achieve flow and increase player retention, it is important that games difficulty matches player skills. Being able to evaluate how people play a game is a crucial component for detecting gamers strategies in video-games. One of the main problems in player strategy detection is whether attributes selected to define strategies correctly detect the actions of the player. In this paper, we will study a Real Time Strategy (RTS) game. In RTS the participants make use of units and structures to secure areas of a map and/or destroy the opponents resources. We will extract real-time information about the players strategies at several gameplays through a Web Platform. After gathering enough information, the model will be evaluated in terms of unsupervised learning (concretely, K-Means). Finally, we will study the similitude between several gameplays where players use different strategies.

Keywords: Player Strategies, Video Games, Sliding Windows, K-Means, Real Time Strategy Game.

1 Introduction

Nowadays a wide number of Computer Science researchers are focused on the develop of intelligence Video Games [1]. Several techniques and methods from areas such as Artificial Intelligence (AI) or Data Mining (DM) have been applied to analyse the gamers behaviours [5], to generate intelligent enemies [13], or to imitate the human behaviour [10], among others. Maybe, one of the most known applications

Fernando Palero · Cristian Ramirez-Atencia · David Camacho
Computer Science Department
Universidad Autónoma de Madrid, Spain
e-mail: {fernando.palero,cristian.ramirez}@inv.uam.es,
david.camacho@uam.es
http://aida.ii.uam.es

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_22

is related to the development of controllers to automatically define real behaviour of Non-Player Characters (NPC). In this topic, there are several works focused on really famous games such as *Ms. PacMan* [6], or *Starcraft* [12]. Other works have been focused on automatic validation levels by finding the different paths that reach the exit [7].

In the literature we find different applications of gamers strategies detection. One interesting application is to study how the gamers interact with the Super Mario Bros game [9] to automatically generate game levels which will enhance player experience. Other researches [15] have proposed a methodology based on feature selection and preference machine learning for constructing models to increase the player satisfaction. In this paper, we present a case study related to the validation of the attributes that model the players strategies based on an *RTS* game - for this research a Tower Defence game, which is a subgenre of RTS game, has been used. This analysis is based on previous works [8] where 4 player strategies, based on unit position distributions, were detected using visualization techniques.

With the strategies identified, the next step is to study their evolution over time and then employ some metric to evaluate whether the attributes modelling them are well defined or not. The metric adopted in this paper to detect these strategies is the similitude between them. To analyse how the player strategies evolve, it is necessary to use a method based on data stream mining. Data stream mining [4] is the process of analysing ordered sequences of data in real-time. A commonly employed technique is sliding windows technique (Section 3.1). Finally, to study the similitude between strategies, a clustering technique (K-Means) is applied [2].

The rest of the paper is structured as follows: section 2 presents a platform for game data extraction and analysis for the *RTS* Game considered. Section 3 describes the methodology used in the data analysis. Section 4 presents some experimental results. Finally, Section 5 shows the conclusions of this research and future lines of work.

2 Web Platform Architecture

This section presents a platform architecture based on a *RTS* game (see Figure 1) that has been developed to study gamers strategies. The platform has been designed using four different modules. The Adaptive Horde Module (*AHM*) is the responsible for generating a fix number of variable hordes of enemies in each wave. The Collector Module (*CM*) allows to automatically extract data from the game, and to gather the interaction from the users. The Strategy Detection Module (*SDM*) analyses the state of the gameplay at the beginning of a new horde and returns a suitable counter-strategy. Finally, the Attribute Validation Module (*AVM*) analyses and returns the distributions of the gameplays. With these distributions, visualization techniques (histograms) have been used to determine the strategies.

AHM receives from *SDM* the parameters necessary to generate the enemies in the different hordes that would appear at each wave. Equation 1 applies the received

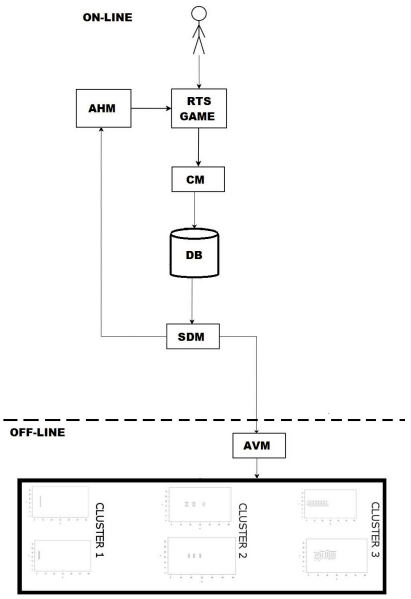


Fig. 1 Framework Architecture based on the RTS game platform

parameters to generate the quantity of different enemies types in the hordes within a gameplay. N represents the number of enemies, β_T the percentage of enemies of type T, W is the current wave and α is a growing factor for the enemies generation. For $\alpha = 0$, the number of enemies in the horde is constant in each wave. If $\alpha = 1$, the amount of enemies grows linearly and, finally, if $\alpha = 2$ the horde has a quadratic growth factor.

$$\#Enemies_T = \beta_T N W^{\alpha} \tag{1}$$

CM extracts data from the RTS game, which in this approach is related to the Unit Data (*UD*). *UD* provides information about the units (in this case towers) based on their features (i.e. position, the unit type, its strength, etc). All this gathered information is then passed to a database module to be stored, and will be recovered in other modules to be analysed.

SDM is responsible for carrying out the analysis of the data gathered by *CM*. It works based on two basic processes: the first process recovers the units information from a data window stored in the database, and the second one calculates both the distributions of X, Y and the euclidean distances from the units to the entry point.

Finally, *AVM*, which works off-line, is responsible for carrying out the validation of the attributes. In a first phase, the distributions that have been calculated in *SDM* are normalized and labelled, and the attributes that help to identify the strategy are obtained from the unsupervised method K-means (see section 3.2). These labels will be used to identify the cluster where the instances have been assigned. Then, in a second phase, the similitude between gameplays strategies is studied (see section 3.3).

3 Description of the Data Analysis Procedure

Three main techniques have been used to achieve the players strategies analysed. The first one is based on a sliding-window technique that is used to gather data and create instances of features. The second one is based on K-means clustering to group distributions by labels. The last technique studies the similitude between the distributions. This section describes these methods.

3.1 Sliding-Window Technique

The most popular approach to deal with data stream involves the use of sliding windows [14]. Sliding-Windows provides a way to divide the data stream into a set of examples to analyse. The procedure for using sliding windows for data stream mining is shown in Algorithm 3. The input of the algorithm is the samples set from the RTS Game. One sample corresponds to one window, and the size of the window is dynamic and changes according to the life time of the wave. The life of a wave is defined as the time between the apparition of its first horde and the disappearance of its last horde. With the size of the windows defined, in each iteration of the algorithm a new window is analysed and the distribution of coordinates X, the distribution of coordinates Y and the distribution of euclidean distances from the units to the exit are returned.

Algorithm 3. This algorithm is an adaptation of [4]

Parameter: $\mathcal{S}$: a data stream of example $\mathcal{W}$: window of examples
Result: $\mathcal{C}$: the distribution of the coordinate X, the distribution of the coordinate Y and the distribution of the euclidean distance from the units to the exit from the window $\mathcal{W}$

```
Initialize window W
forall the example x_i ∈ S do
    W ← W ∪ {x_i}
    build C using W
end
```

3.2 K-Means Clustering

K-means algorithm is used to partition the input data set into k partitions. However, K-Means algorithm has two problems. The first one, in contrast to other algorithms, is that K-Means cannot be used with arbitrary distance functions or on non-numerical data. And the second one, K-Means algorithm cannot guarantee finding the best space partition. To solve the first problem we use the euclidean distance and transform all dataset to numerical data. For the second, we execute the algorithm

using always the same *'k'* several times (see Algorithm 4) and then we choose the best result returned. For this selection, we use two metrics: intra-cluster distance and inter-cluster distance. Intra-cluster [11] distance measures (equation 2) the average of the distances between the points and its respective cluster centroids. In the equation 2 we can see the intra-cluster metre, where N is the number of instances of data extracted from the game, K is the number of clusters, and z_i is the centroid of cluster C_i.

$$intra(x, z_i) = \frac{1}{N} \sum_{i=1}^{K} \sum_{x \in C_i} |0x - z_i|0^2 \tag{2}$$

$$inter(z_i, z_j) = min|0z_i - z_j|0^2; i = 1, 2, ..., K-1; j = i+1, ..., K; \tag{3}$$

Inter-cluster distance (equation 3) measures the distance between cluster centres. To choose the result that makes a good partition of the data space, it is necessary to minimize the intra-cluster distance and maximize the inter-cluster [3] distance measure. The aim is to minimize the validity measure (equation 4).

$$validity(x, z_i, z_j) = \frac{intra(x, z_i)}{inter(z_i, z_j)} \tag{4}$$

Algorithm 4. Algorithm to choose the best K-Means partitioning

Parameter: $\mathcal{W}$: window of examples
Result: $\mathcal{C}$: data labelled in the window $\mathcal{W}$

```
Initialize window W
k=4
vecValidity ← []
for j = 1 to 10 do
    labels ← KMeans(k, W)
    validity ← CalculateValidity(labels, W)
    vecValidity(i) ← validity
end
vecK(k) ← min(vecValidity)
```

3.3 Distribution Similitude

The different $\mathcal{W}$ that compose the strategies are labelled by groups with K-Means to study the similitude between distributions. Equation 5 represents the similitude between two distributions D_1 and D_2. In this equation, w_i represents the wave number, and $D_1(w_i)$ and $D_2(w_i)$ indicate the group of the distribution in w_i. The similitude is calculated dividing the number of the label coincidences from the distributions ($D_1(w_i)$ and $D_2(w_i)$) between the number of waves.

$$Similitude(D_1, D_2) = \frac{\#\{i \epsilon \{1 \dots \#Waves\} | D_1(w_i) = D_2(w_i)\}}{\#Waves} \quad (5)$$

4 Experimental Results

We have studied the similitude between the strategies detected in the previous work [8] (*Zigzag*, *Horizontal*, *Grouped* and *Vertical* distributions). In this new approach, we are interested in calculating the similitude between strategies. For this purpose, we have previously analysed and labelled the gameplays dataset according to the strategies identified. With this experiment we determine if the features selected to distinguish the strategies are adequate.

4.1 Distributions Similitude Comparison

Sliding-window technique has been used to extract the ten wave distribution of a gameplay. The distributions are grouped by labels that are assigned by K-Means algorithm. We choose a $k = 4$ for K-Means that corresponds to the number of strategies found in the previous work. Finally, we use these labelled groups to study the similitude between strategies distributions, using equation 5 for this purpose. With these groups we have obtained a table of similitude (table 1).

Table 1 Similitude between distributions, where Z is the Zigzag distribution, G is the Grouped distribution, H is the Horizontal distribution and V is the Vertical distribution.

Game ID		1	2	3	4	5	6	7	8	9	10	11	12
	Distribution	Z	Z	Z	G	G	G	H	H	H	V	V	V
1	Z	1	**0,6**	**0,1**	0,1	0	0,5	0,6	**0,5**	0,5	0,5	**0,5**	0,5
2	Z	0,6	1	0	0,2	0	0,4	0,4	0,4	0,4	0,4	0,4	0,4
3	Z	0,1	0	1	0,1	0,1	0,1	0	0	0,1	0	0	0,1
4	G	0,1	**0,2**	0,1	1	**0,7**	0,3	**0,2**	0,2	0,3	0,1	0,1	**0,3**
5	G	0	0	0,1	0,7	1	0,1	0	0	0,1	0	0	0,1
6	G	0,5	0,4	0,1	0,3	0,1	1	0,9	0,9	1	0,8	0,8	1
7	H	0,6	0,4	0	0,2	0	**0,9**	**1**	**0,9**	**0,9**	**0,8**	**0,8**	**0,9**
8	H	0,5	0,4	0	0,2	0	0,9	0,9	1	0,9	0,8	0,8	0,9
9	H	0,5	0,4	0,1	0,3	0,1	1	0,9	0,9	1	0,8	0,8	1
10	V	0,5	0,4	0	0,1	0	0,8	0,8	0,8	0,8	1	1	0,8
11	V	0,5	0,4	0	0,1	0	0,8	0,8	0,8	0,8	1	1	0,8
12	V	0,5	0,4	0,1	0,3	0,1	1	0,9	0,9	1	0,8	0,8	1

In table 1, it is appreciated that similitude between different gampeplays using Zigzag distribution is low. The same happens with Grouped distributions. This means that these kind of strategies are difficult to identify. Moreover, different distributions could be included inside this strategy, so it is necessary to take more samples. On the other hand, the similitude between gameplays using Horizontal or Vertical distributions is high. This implies that these strategies are well identified by the employed attributes (units positions distributions).

Looking at the similitude between different strategies, we observe that Zigzag distributions can be confused with other distributions, as the similitude values between two gameplays using Zigzag distributions are similar to those using a Zigzag distribution and other type of distribution. This confirms the aforementioned inclusion of strategies inside Zigzag distributions. For Grouped distributions, the same problem appears.

Finally, comparing the similitude of gameplays using Horizontal and Vertical distributions, we can appreciate that these similitude values are high in all cases.This happens because the metric that we use to generate the features does not consider the orientation of the distribution. We can observe that Vertical distribution is the inverse of Horizontal distribution. In Vertical distribution, we can see that the unit distribution along the axis X is one bin. However, in Horizontal distribution, we have this behaviour in the Y axis.

In conclusion, we find that the attributes based on the units positions distributions is not very effective to identify player strategies. We only can distinguish between linear (*Horizontal* or *Vertical*) distribution or not linear. It is necessary to use more attributes that help to improve this model.

5 Conclusions and Future Work

This work provides an evaluation of the attributes used to identify strategies in gameplays. To achieve this purpose, a framework based on an RTS platform has been designed, and the strategies detected in a previous work based on the units positions distributions have been employed. The experiment carried out tries to determine if the attributes selected to identify the players strategies are sufficiently descriptive. For this purpose, we have used K-means to group strategies. Latter these groups have been used to calculate the similitude between gameplays.

From the similitude study, we have concluded that the attributes employed to detect players strategies have low performance. We have found that the actual attributes identifies linear distributions (*Horizontal* or *Vertical*), but they are not good at discriminating not linear distributions (*Grouped* or *Zigzag*), due to it is necessary to use more attributes.

In future works, it will be necessary to use more features to perform a better classification and study more unsupervised techniques, such as spectral clustering, to determine which do a best data partitioning. Moreover, we could apply Online Learning, so the attributes are updated every time new data is gathered.

References

1. Alayed, H., Frangoudes, F., Neuman, C.: Behavioral-based cheating detection in online first person shooters using machine learning techniques. In: 2013 IEEE Conference on Computational Intelligence in Games (CIG), pp. 1–8. IEEE (2013)
2. Alsabti, K., Ranka, S., Singh, V.: An efficient k-means clustering algorithm. Association for the Advancement of Artificial Intelligence (1997)
3. Bello-Orgaz, G., Menendez, H., Camacho, D.: Adaptive k-means algorithm for overlapped graph clustering. International Journal of Neural Systems 22(05), 1–19 (2012)
4. Brzezinski, D.: Mining Data Streams with Concept Drift. Master's thesis, Poznan University of Technology (2010)
5. Dey, R., Child, C.: QL-BT: Enhancing behaviour tree design and implementation with Q-learning. In: 2013 IEEE Conference on Computational Intelligence in Games (CIG), pp. 1–8. IEEE (2013)
6. Gagne, D.J., Congdon, C.B.: Fright: A flexible rule-based intelligent ghost team for Ms. Pac-Man. In: 2012 IEEE Conference on Computational Intelligence and Games (CIG), pp. 273–280. IEEE (2012)
7. Gonzalez-Pardo, A., Palero, F., Camacho, D.: An empirical study on collective intelligence algorithms for vide games problem-solving. Computing and Informatics (In press, 2014)
8. Palero, F., Gonzalez-Pardo, A., Camacho, D.: Simple Gamer Interaction Analysis through Tower Defence Games (submited, 2014)
9. Pedersen, C., Togelius, J., Yannakakis, G.N.: Modeling player experience in Super Mario Bros. In: IEEE Symposium on Computational Intelligence and Games, CIG 2009, pp. 132–139. IEEE (2009)
10. Polceanu, M.: MirrorBot: Using human-inspired mirroring behavior to pass a Turing test. In: 2013 IEEE Conference on Computational Intelligence in Games (CIG), pp. 1–8. IEEE (2013)
11. Ray, S., Turi, R.H.: Determination of number of clusters in k-means clustering and application in colour image segmentation. In: Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques, pp. 137–143 (1999)
12. Synnaeve, G., Bessiere, P.: A Bayesian model for RTS units control applied to StarCraft. In: 2011 IEEE Conference on Computational Intelligence and Games (CIG), pp. 190–196. IEEE (2011)
13. Traish, J.M., Tulip, J.R.: Towards adaptive online RTS AI with NEAT. In: 2012 IEEE Conference on Computational Intelligence and Games (CIG), pp. 430–437. IEEE (2012)
14. Wang, H., Fan, W., Yu, P.S., Han, J.: Mining concept-drifting data streams using ensemble classifiers. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2003, pp. 226–235. ACM, New York (2003)
15. Yannakakis, G.N., Hallam, J.: Feature selection for capturing the experience of fun. In: Proceedings of the Artificial Intelligence and Interactive Digital Entertainment, vol. 7, pp. 37–42 (2007)

Part VII
Linked, Open and Big Data

Semantic Information Fusion of Linked Open Data and Social Big Data for the Creation of an Extended Corporate CRM Database

Ana I. Torre-Bastida, Esther Villar-Rodriguez,
Javier Del Ser, and Sergio Gil-Lopez

Abstract. The amount of on-line available open information from heterogeneous sources and domains is growing at an extremely fast pace, and constitutes an important knowledge base for the consideration of industries and companies. In this context, two relevant data providers can be highlighted: the "Linked Open Data" and "Social Media" paradigms. The fusion of these data sources – structured the former, and raw data the latter –, along with the information contained in structured corporate databases within the organizations themselves, may unveil significant business opportunities and competitive advantage to those who are able to understand and leverage their value. In this paper, we present a use case that represents the creation of an existing and potential customer knowledge base, exploiting social and linked open data based on which any given organization might infer valuable information as a support for decision making. In order to achieve this a solution based on the synergy of big data and semantic technologies will be designed and developed. The first will be used to implement the tasks of collection and initial data fusion based on natural language processing techniques, whereas the latter will perform semantic aggregation, persistence, reasoning and retrieval of information, as well as the triggering of alerts over the semantized information.

Keywords: Big Data, Social Media, Linked Open Data, business intelligent, information fusion, ontology management, information modelling.

1 Introduction and Motivation

Nowadays, organizations need to gather valuable information that will allow them to improve their business processes and optimize their decision making. In this context,

Ana I. Torre-Bastida · Esther Villar-Rodriguez · Javier Del Ser · Sergio Gil-Lopez
TECNALIA, OPTIMA Unit, E-48160 Derio, Spain
e-mail: {isabel.torre,esther.villar,javier.delser,
sergio.gil}@tecnalia.com

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_23

business intelligence [1] is the set of strategies, relevant aspects and key technologies to the creation of knowledge on the data environment, through the analysis of these and the context, with the ultimate aim to facilitate business decision making. However, the principal problem to achieve this task is the vast amount of available data and the efficient extraction of useful information from huge repositories. The problems associated with data volume is the concept known as Big Data, where the collection of data sets is so tremendous and complex that their processing using traditional data management tools results computationally unaffordable. Two of the most notable data providers in Big data are the Linked Open Data (LOD [2]) and social big data. Social media is becoming an important context-rich information source for organizations and therefore many business executive consider an essential challenge to be faced in order to incorporate this user-generated information in their decision-making chain. The goal is that businesses achieve profits from social platforms such as Wikipedia (DBpedia in the LOD), Facebook or Twitter. Due to the heterogeneity of the received digitized data, following non-standard schemas and with low accuracy and reliability, a great human effort becomes necessary to extract, format and assimilate, trying to solve the second major problem, which is the removal of noise in data content before using it. A third problem arising therefrom is how to get to merge these datasets with traditional business data, such as relational database or corporate knowledge systems.

Many projects follow these business intelligence research lines in areas such as brand recognition, competitor analysis or benchmarking [3–5]. But there are scarce studies applying them to a specific matter such as customer relationship management towards potential customers identification or existing clients' information improvement or enrichment. Aimed at filling this gap, our approach defines a system capable of 1) implementing the generation and management of an extended corporate CRM database; 2) solving several related analytics problems (knowledge discovering and aggregation/fusion) stemming therefrom; and 3) exploiting emerging data sources by using the semantic and big-data technology stack. Technically, our system follows a semantic aggregation approach that allows taking advantage of the LOD datasets structure so as to enhance our solution. In detail, the main contributions of our scheme are the following:

- Analysis of the particular business problem of discovering and improving organizations customer database.
- Exploitation of new data sources (social media, LOD).
- Making use of the semantic and big-data technology stack for data collection and aggregation tasks.

In the rest of this paper we first introduce the main concepts related to our approach, Social Media, Linked Open Data and the closest related work. Then our scheme and its core processing steps are described in detail. Finally, we illustrate a study use case to evaluate our system prototype.

2 Background

The web has recently undergone a transformation in the amount and type of available contents, emerging a new paradigm called Big Data. This new term is used to describe the exponential growth and availability of data, both structured and unstructured. In our approach we use two clear examples of these kind of datasets: social big data (unstructured) and Linked Open Data (semantically structured).

Nowadays social media platforms are storing enormous amounts of no previously automatically analyzed data that could reveal critical information. The reason behind is that the user role has shifted from being a mere consumer to a content provider. Social media is defined by Kaplan et al. [7] as"a group of Internet-based applications that build on the ideological and technological foundations of Web 2.0, and that allow the creation and exchange of user-generated content". For this reason it can be considered as a context-rich source of big data and is usually referred to as Social Big Data. To better explain this definition we must introduce two concepts: *Web 2.0* and *User Generated Content* (UGC). Web 2.0 describes a new method in which software developers and end-users collaborate on the World Wide Web; that is, content and applications are no more statically published by an individual, but are continuously modified by a collaborative users community instead. UGC describes the various forms of media content that are publicly available and created by end-users.

On the other hand, Linked Data refers to a set of best practices for publishing and interlinking structured data on the Web. With this definition, Bizer et al. [6] defined the linked data paradigm and provided a mechanism for building the Web of Data, what is based on the semantic Web technologies and it may be considered as a simplified version of the Semantic Web. The data model for representing interlinked data is RDF [3], where data is represented as node-and-edge-labeled directed graphs. Some published Linked Data sets contain billions of triples, whose cardinality is steadily increasing to yield the Linked Data Cloud, i.e. a group of data sets available on the Web as Linked Data with links pointing at other Linked Data sets.

2.1 Related Work

In business intelligence – especially in the area of competitive information (CI) – the data gathering process can involve a large number of research areas regarding to technologies and strategies, which have unchained an intense activity in the related literature. Our approach deals with social big data which has been broadly adopted to nourish data analytic systems. Studies as the one by Rappaport in [8] introduces the essential role it can take to exploit social media in the field of business intelligence, presenting cases of study in which the social media data is turned into business advantages. In [9] a preliminary study about using text-mining techniques in the task of collaborative intelligence information gathering is presented. The main difference with our approach lies on the used techniques (our work resorts to

semantic fusion and big data technologies) and the application domain, which in our case lays on the specific example of creating a knowledge base of customers. Another work related to our scheme is the one presented by Shroff et al. in [10], which describes a framework for the fusion of business intelligence in various industries such as manufacturing, retail or insurance. Once again and unlike our proposal, this contribution hinges on its global and general case-based implementation without concentrating on a specific problem. Furthermore, the artificial intelligence techniques used in their work are the blackboard architecture and the locality sensitive hashing, which are far away from our semantic fusion approach. Another interesting and more specific work is presented by Agichtein et al. in [11], which elaborates on a high-quality social media information gathering scheme, but only managing data posted in *Yahoo!Answers* social platform. Other investigations also discuss the advantages of data fusion on information collected from social media as in [12], in which multiple features in the social media environment (textual, visual and user information) are fused for later being used on a retrieval algorithm for large social media data (*flickr*). Likewise, in [13] a use case of a shared on-line calendar is presented and enhanced with events generated by user social networks and location data using fusion techniques. Further, we highlight the work presented by Kim et al. in [14] due to the fact that it is the only one that uses semantic fusion techniques. However, its purpose is out of the scope of business intelligence and does not provide enough technical details. Its methodology and assessment is deemed as insufficient for a fair comparison with the technical proposal next presented. Finally, we analyze the work done by Hoang et al. in [15], where a survey about technologies and applications of Linked data mahups as well as the challenges to build them are presented. In this paper a use case close to our approach is presented, since both use semantic technologies for integration. The main difference lies in the architecture (they use semantic web pipes and our approach instead uses ontology mapping/alignment techniques for the semantic integration) and datasets (they exploit freebase and do not use social media data sources).

3 Information Collection from the Web

Our system collects external information, such as company related tweets, customer feedback (comments) or business related open data and merges this data with traditional enterprise databases, with the aim at storing these aggregated information following an adequate business semantic model. This section delves into the first of these tasks, the information collection from two different on-line available sources: the Social Media and Linked Open Data Cloud, as well as into the subsequent filters to extract their relevance:

1. Social Big Data: at this point the data collected in two streaming social platforms is selected.
   a. Facebook posts from specific user-ids are the considered data, extracting the comments generated by these users.
   b. Twitter feeds containing certain keywords or from specific user-ids. These keywords are extracted using a TF/IDF approach from the corporation documents and website.

   The technology used to perform these tasks are Facebook[1] and Twitter[2] streaming application programming interfaces (API).
2. Linked Open Data Cloud: there are several datasets related to the business domain, such as DBpedia, CrunchBase or Freebase, which can be queried by the SPARQL query language or web services. From these datasets, structured information about customers is obtained, which is latter mapped to the semantic model of our system.

The data collection is detailed in Figure 1. This task is composed by three subprocesses: data collection and noise reduction, extraction of disambiguated entities and harvesting of related entities information available as open data.

In a first step, the different social media data streams are captured using the aforementioned APIs. Next, the posts(from Twitter or Facebook platforms) are preprocessed. At this stage we use the Freeling API[3] to carry out the language analysis, calculating their corresponding synsets (i.e. a group of data elements that are considered to be semantically equivalent, represented by an identifier). The collection of pairs formed by each post and its synsets are the input events to a set of rules that allow deducing if the post (tweet/comment) can be considered within the business domain. This stage is what we have coined as NOISE filter. This filter, composed by a set of rules is implemented by the Esper CEP engine[4] built up from a set of synsets constructing a context that describes and models the business domain, for example `concept: business; synsets:08056231,08058098`. If any of the synsets belonging to ongoing post can be match to any one of the synsets that form the context, the rule is activated and the post is filtered as pertinent. Otherwise the post is discarded since it is assumed that its content is not about anything related to the business world.

Once posts have gone through the noise filter, the result will be deemed valuable since it is likely to provide meaningful information about the domain, which is then fed to the subprocess in charge of entity extraction. Named Entity Recognition (NER) refers to the module or function in charge of detecting any kind of entities such as cities, organizations, people and is mostly utilized by NLP utilities as a contributor for semantic information. In our case, the filtered posts can contain

[1] `https://developers.facebook.com/docs/graph-api`
[2] `https://dev.twitter.com/docs/api/streaming`
[3] `http://nlp.lsi.upc.edu/freeling/`
[4] `http://esper.codehaus.org/`

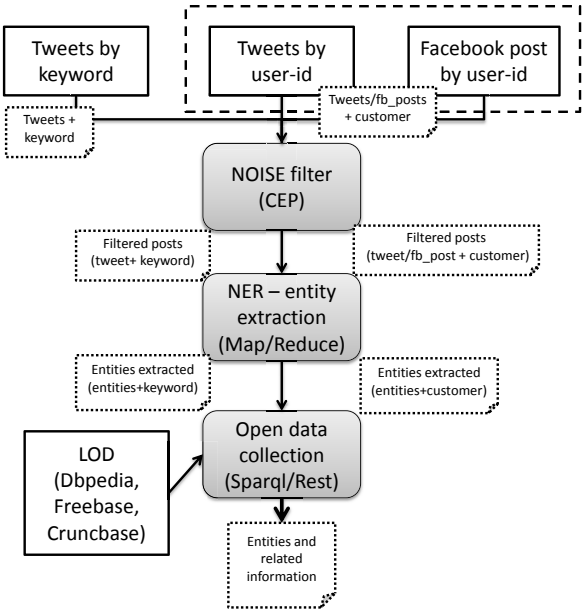


Fig. 1 Data collection process flow

any named entity corresponding to an already existing customer, a potential client or even a competitor working in the same market sectors. On this purpose, Daedalus Topic Extraction API has been selected, integrating it on a Map-Reduce framework to parallelize the algorithm responsible for extracting entities. The output obtained from the Map-Reduce job is a set of entities grouped by post. Finally, for each of the previously extracted entities, we will collect the information available in the Linked Open Data sets (freebase, DBpedia) and other open data sets such as CrunchBase. This information will be merged and aggregated to the existing data from corporation relational databases, with the final aim of feeding the semantic model.

4 Semantic Fusion: Aggregation, Model and Interlinking

The semantic aggregation process has two main goals: to improve the existing information for customers of the organization and to discover new potential customers. The entire process is detailed in Figure 2. First of all a classification process is applied to each post to determine whether its contents relate to any entity existing in the semantic data model. Depending on the result of this classification the system follows two different alternative flows. In the positive case, the semantic model is updated with the new information about customer and its partnerships/relationships. Otherwise, the data gathered from the Linked Open Data Cloud is mapped into a

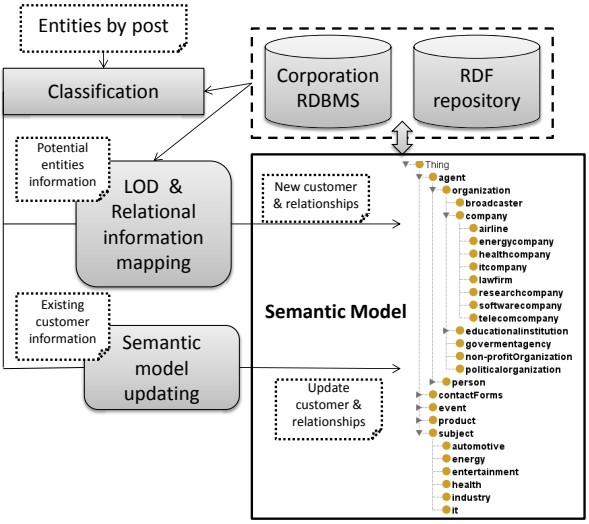


Fig. 2 Semantic data aggregation

new instance within the semantic model. These processes are supported by a set of previously computed semantic links between our model and the LOD datasets vocabularies, which are calculated following the ontology alignment process proposed by the authors in [16].

With regard to the definition of our model schema, well-known semantic vocabularies will be reused, to promote interoperability with other RDF repositories or datasets. Our ontology model is based on the combination of the `schema.org` ontology along with that used in DBpedia and vocabularies as SKOS to specify semantic relationships and links. New classes or properties are also modeled in the case that existing vocabularies do not provide their definition. Finally, the new instances of the semantic data model are stored in the Virtuoso Open-Link RDF repository [5].

5 Information Retrieval, Inference and Alert Generation

Once the information has been converted to RDF format following our semantic model and it is saved in the RDF repository, some added-value operations can be implemented over the stored data, such as the following features:

- Information retrieval: In our case SPARQL – the current W3C recommendation for querying RDF data – is selected to allow users to perform selective queries. In

[5] `http://virtuoso.openlinksw.com/`

our system the SPARQL endpoint provided by RDF repository Virtuoso OpenLink and the JENA API[6] are the chosen tools for implementing this module.

- Inference: Based on the information stored in the repository semantic inference processes (RDFS and OWL) can be performed with the aim of discovering new relationships. This task can be accomplished by semantic reasoners like Pellet [17], in combination with the JENA API. This process also allows for the definition of specific business semantic rules implemented using the SWRL (*Semantic Web Rule Language combining OWL and RuleML*) language.
- Alert generation: Finally, an alert generation module is responsible for monitoring the data and triggering events that indicate that a number of conditions specified in the alert have been fulfilled. For its implementation a listener is utilized during the loading and inference process that allows detecting whether alert conditions have been met.

6 Use Case

This section describes in detail an illustrative example of the process followed by our system since the data collection occurs until the information is retrieved by a

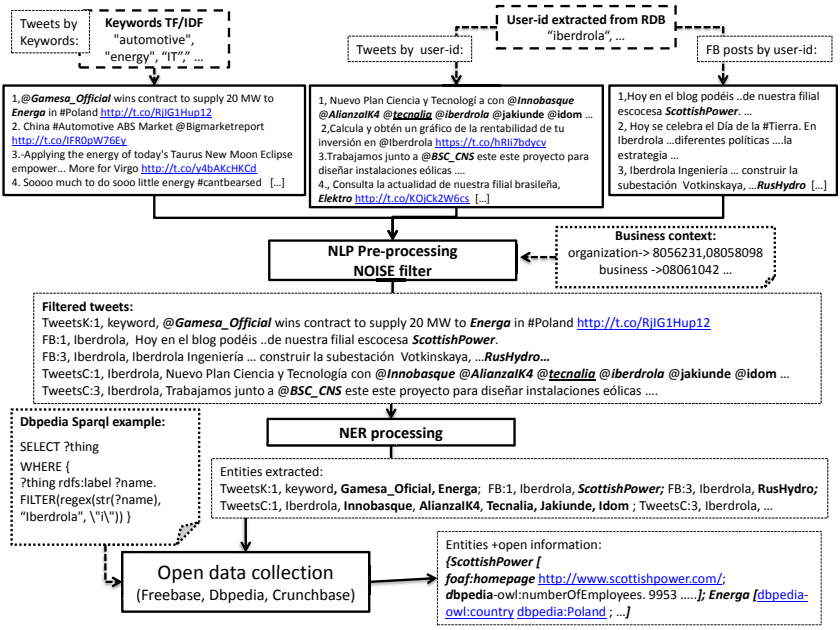


Fig. 3 Data collection example

[6] https://jena.apache.org/documentation/query/

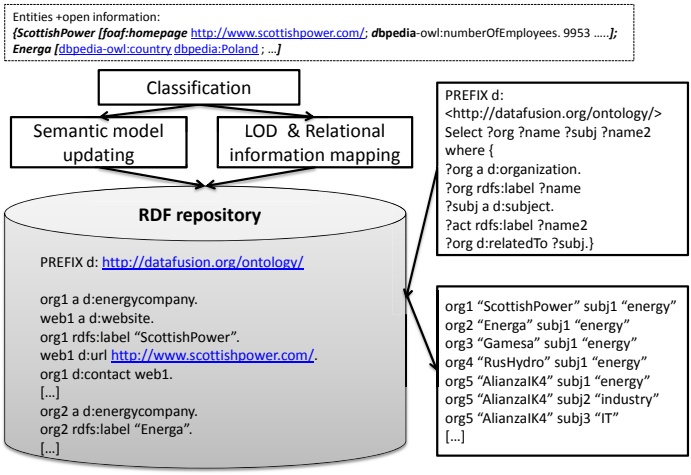


Fig. 4 Generated semantic data model and Sparql execution example

SPARQL query. The data collection process is shown in Figure 3. The first input is the *real* data retrieved from Twitter and Facebook. Tweets and posts are preprocessed to transform them in synsets as explained in Section 3. These synsets are filtered (a noise filter for irrelevant data) using a macro that consists of a set of synsets representing the business domain (business context in the Figure). These filtered tweets and posts are subject to a named entity recognition procedure aimed at extracting the entities so as to collect from them the information available on the LOD.

Finally, the data model and instances generated by the semantic aggregation process and an example of information retrieval using a SPARQL sentence are shown in Figure 4. As shown in the picture, the query returns a list of all organizations and its related subjects. In this context it must be noted that although `ScottishPower` is annotated as `energycompany`, this entity is also returned in the query, because in the ontological model (see figure 2) an `energycompany` is categorized as a subclass of `organization`. This unveils one of the advantages of using a semantic model for information retrieval.

7 Concluding Remarks and Future Research

This manuscript has gravitated on the problem of automatically creating and managing a customer database from a novel perspective: semantic aggregation. Input data comes from new sources such as social media and Linked Open Data. Furthermore, different modules have been implemented leveraging Big Data (Map-Reduce,

Complex Event Processing) and semantic web (RDF repository, reasoner, SWRL) technology stacks. A use case exemplifies the multiple possibilities and potentiality offered to a corporation by our approach, ranging from the discovery of new customers to the knowledge base expansion of traditional clients. This springs profitable advantages in the business domain, where the decision making is a critical process and the collection of customer information is a key factor.

Future work will be devoted towards the study of new applications and enlarging the technical scope of this semantic aggregation so as to e.g. also include projects referencing entities, business concepts or places and properties that can be matched to relationships to the model inferred from the posts thanks to developing new algorithms that use PLN and classification techniques. Furthermore multilingual features will be also considered for their inclusion in the platform.

References

1. Moss, L.T., Atre, S.: Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications. Addison-Wesley (2003)
2. Bizer, C., Heath, T., Idehen, K., Berners-Lee, T.: Linked data on the web (LDOW2008). In: Proceedings of the 17th International Conference on World Wide Web, pp. 1265–1266 (2008)
3. Hoffman, D.L., Fodor, M.: Can you measure the ROI of your social media marketing. MIT Sloan Management Review 52(1), 41–49 (2010)
4. Vuori, V.: Social media changing the competitive intelligence process: elicitation of employees' competitive knowledge. Tampereen teknillinen yliopisto. Julkaisu-Tampere University of Technology. Publication; 1001 (2011)
5. Bingham, T., Conner, M.: The new social learning: A guide to transforming organizations through social media. Berrett-Koehler Publishers (2010)
6. Bizer, C., Heath, T., Berners-Lee, T.: Linked data-the story so far. International Journal on Semantic Web and Information Systems 5(3), 1–22 (2009)
7. Kaplan, A.M., Haenlein, M.: Users of the world, unite! The challenges and opportunities of Social Media. Business Horizons 53(1), 59–68 (2010)
8. Rappaport, S.D.: Listen First!: Turning Social Media Conversations Into Business Advantage. John Wiley and Sons (2011)
9. Dey, L., Haque, S.M., Khurdiya, A., Shroff, G.: Acquiring competitive intelligence from social media. In: Proceedings of the 2011 Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data, p. 3. ACM (2011)
10. Shroff, G., Agarwal, P., Dey, L.: Enterprise information fusion for real-time business intelligence. In: IEEE International Conference on Information Fusion (FUSION), pp. 1–8 (2011)
11. Agichtein, E., Castillo, C., Donato, D., Gionis, A., Mishne, G.: Finding high-quality content in social media. In: Proceedings of the 2008 International Conference on Web Search and Data Mining, pp. 183–194. ACM (2008)
12. Cui, B., Tung, A.K., Zhang, C., Zhao, Z.: Multiple feature fusion for social media applications. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, pp. 435–446. ACM (2010)
13. Lovett, T., O'Neill, E., Irwin, J., Pollington, D.: The calendar as a sensor: analysis and improvement using data fusion with social networks and location. In: Proceedings of the 12th ACM International Conference on Ubiquitous Computing, pp. 3–12. ACM (2010)
14. Kim, H., Son, J., Jang, K.: Semantic Data Fusion: from Open Data to Linked Data. In: Proceedings of the European Semantic Web Conference (2013)

15. Hanh, H.H., Tai, N.C., Duy, K.T., Dosam, H., Jason, J.J.: Semantic Information Integration with Linked Data Mashups Approaches. International Journal of Distributed Sensor Networks 2014, Article ID 813875 (2014)
16. Torre-Bastida, A.I., Villar-Rodriguez, E., Del Ser, J., Camacho, D., Gonzalez-Rodriguez, M.: On Interlinking Linked Data Sources by Using Ontology Matching Techniques and the Map-Reduce Framework. In: Corchado, E., Yin, H. (eds.) IDEAL 2014. LNCS, vol. 8669, pp. 53–60. Springer, Heidelberg (2014)
17. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical owl-dl reasoner. Web Semantics: Science, Services and Agents on the World Wide Web 5(2), 51–53 (2007)

Time-Frequency Social Data Analytics for Understanding Social Big Data

Duc T. Nguyen, Dosam Hwang, and Jason J. Jung*

Abstract. Social Network Services (SNS) have been the most popular channel where users can generate and disseminate a large amount of information (so-called 'social big data') among other users efficiently. Discovering meaningful patterns from these SNS (e.g., clustering relevant messages, detecting events, and understanding trends of social communities) is an important, but difficult research issue on social big data analytics. In this paper, we present an on-going work to transform social data in time domain to in frequency domain for detecting meaningful events from the social big data. Consequently, this work is expected to significantly reduce the volume (and also, complexity) of the social data and to improve the performance of the data analytics.

1 Introduction

In this paper, we focus on how to detect social events [1,7] from social data streams on SNS. The dataset usually consist of (i) microposts (i.e., the data which has been generated by users on SNS), (ii) timestamps (i.e., when the microposts have been generated), (iii) and a social structure of the users. The main goal is to solve a special challenge of detecting what are happening/most discussing on SNS by considering an event is a subject which strongly draw attention from users. The evidence for

Duc T. Nguyen · Dosam Hwang
Yeungnam University, 712-749, Korea
e-mail: {duc.nguyentrung,dosamhwang}@gmail.com

Jason J. Jung
Chung-Ang University, 156-756, Korea
e-mail: j2jung@gmail.com

* Corresponding author.

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_24

an event's occurrence is that users start to discuss frequently/emphatically and keep talking about the topic and its relevant things in a period called life-cycle of event. For event detection purpose, we convert all the collected data into signals of individual terms (i.e. words) instead of storing full raw microposts. Each signal represents a distribution of term's scores over time. It is the raw material for extracting event in the Algorithm 5.

2 Related Work

In [3, 6], authors attempt to analyze the social data by using preprocessing methods which transforms text-based data into different dimensional measurements. He et al. [3] analyze the signals of words in the collected microposts, where each signal is samples of number of word occurrences over time, an analyzing method in frequency domain is applied for detecting event from the given corpus. Basically, the event detection process is done by finding peaks in frequency domain; determining category of power spectrum strength and periodicity of signals. This method is extended in a research of Weng, et al. [6], where authors suggest to use wavelet transformation to keep time information and frequency together as the result of preprocessing tasks. They assumed that several words are used more frequently when an event occurs, the number of its occurrence is increased rapidly in a short step, these words are called 'burst' words. Only signals of burst words are captured, its cross-correlation degree is used to find subgroups of strong relevant words by using modularity-based graph partitioning. Each group of words is predefined as an event with corresponded statuses in the corpus.

3 System Modelling

3.1 Data Representation

In this work we consider an event in Social Data is a subject which continuously attracts a large number of discussion in a short time period. So that, the distribution of the microposts in that discussion has strong intensity at some certain positions, which is a sign for detecting the event's occurrences. We demote $T = [t_1 : t_2]$ is the time period of the collected data; ΔT is the time interval between two samples, the time unit can be specified as minutes, hours or days. A small value ΔT make to determine event life-cycle more accurately, while in opposite a larger value can reduce the storage size for saving signals. Value of ΔT also affects to the total amount of time needed for processing, so it should be selected in the balance between the two demands. $N = (t_2 - t_1)/\Delta T$ is the number of samples in time domain of signals. A signal of word w in time domain is described as

$$w(i) = df.idf_w(i),\ \forall i \in [0 : N) \tag{1}$$

where N is an exponential value of 2. The value $df.idf_w(i)$ is the weight of w at the sample i^{th}, it is defined by extending tf.idf score [4] as

$$df.idf_w(i) = \frac{DF_w(i)}{M(i)} \times \log \frac{M}{DF_w} \tag{2}$$

where $DF_w(i)$, $M(i)$ respectively are the total number of microposts containing keyword w and the total number of microposts collected at the sample i^{th}. DF_w is total number of microposts containing w and M is total number of microposts in the given corpus. Figure 1(a) is an example of the signal of keyword 'Goal'.

3.2 Detecting Events

Assuming that an event or topic can not be summarized by one word, but is a set of meaningful keywords in the list of diffused microposts on SNS. We denote KW is a set of unique keywords in the given corpus, $CW_e \in KW$ is a subset of KW which strongly expresses an event e. Each event has an unique set of keywords called characteristic set, two characteristic sets are different in at least one element.

Definition 1 (Event signal). Event signal is a combination from signals of each keywords in its characteristic set. It is represented as

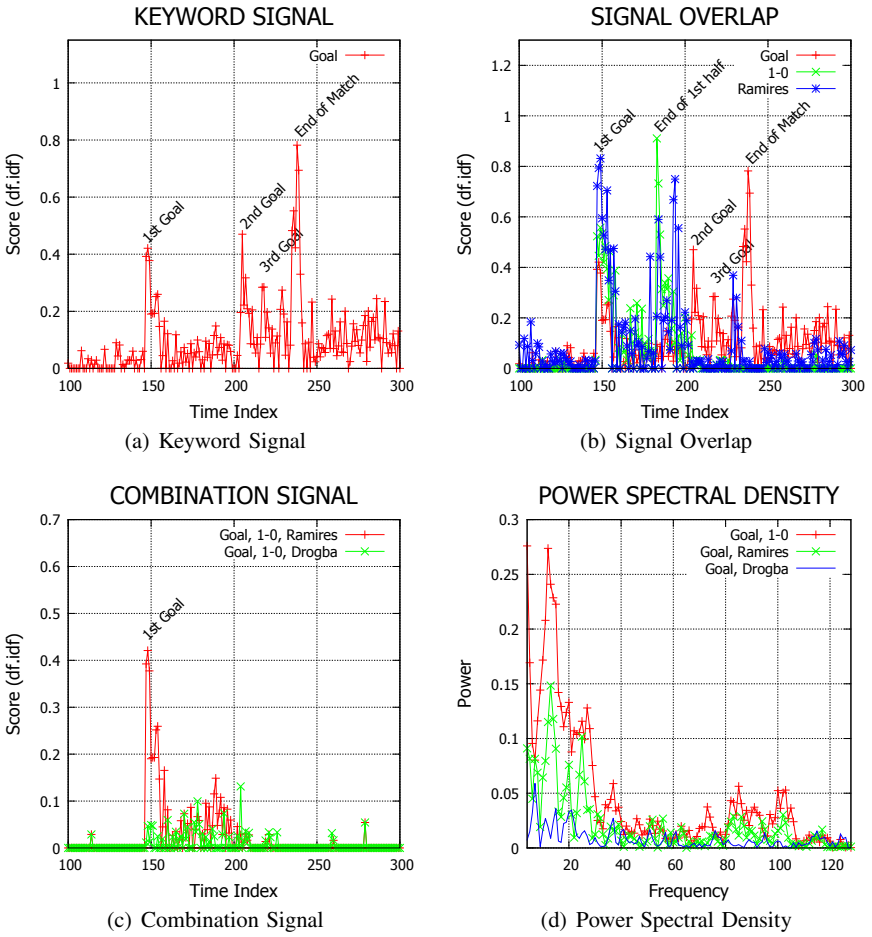
$$e(i) = \min_{\forall w \in CE_e} w(i),\ \forall i \in [0 : N) \tag{3}$$

Its power spectral density is calculated as

$$PSD_e(x) = \frac{1}{N}|E(x)|^2,\ \forall x \in [0 : \frac{N}{2}] \tag{4}$$

where $E(x)$ is the FFT series of $e(i)$.

With above definition, an event signal is a minimization of keyword signals at each sample points. The power of combination signal reaches to a stable value maintained around the timestamps of event's occurrences. Obviously it is clear, the power of combination signal is decreased if its characteristic set CW_e contains any irrelevant keywords. Because inconsistent signals attenuates the final signal at some certain points. Figure 1(b) shows an overlap between signals of keywords 'Goal', '1-0', and 'Ramires' at the timestamp of the first goal in the match between Chelsea and Liverpool in FA Cup 2012, which is scored by Ramires at 11^{st} minute. And the combination of the relevant signals maintains its power around the event occurrence's timestamp, while the rest case losses its power as shown in Figure 1(c). In the example, Drogba is the Chelsea player who scored the second goal of match, hence the signal of keyword '1-0' is the main factor to weaken the combination signal of the keyword set {Goal, 1-0, Drogba}.



(a) Keyword Signal (b) Signal Overlap

(c) Combination Signal (d) Power Spectral Density

Fig. 1 The demonstration of micropost signals collected around the final football match between Chelsea and Liverpool in The FA Cup 2012 (a) The signal of keyword 'Goal'; (b) The overlap of keyword signals related to the 1st goal of match; (c) The combination signals of two keyword sets {Goal, 1-0, Ramires} and {Goal, 1-0, Drogba}; (d) The Power Spectral Density of combination signals of three keyword set s_1 = {Goal, 1-0}, s_2 = {Goal, Ramires} and s_3 = {Goal, Drogba}

For analysis the signals in frequency domain, we use Power Spectral Density (PSD) concept, which is a distribution of the total signal power over frequencies, to measure how strong of the signal per an unit bandwidth of frequency interval. We assume that the topic's keywords distribute with correlated frequencies even if it is delayed a short time period. Hence the PSD of signal give us a chance for detecting what keywords are oscillated together. Let consider the example in Figure 1(d), the combination signals of keyword sets have different PSD oscillation patterns. While signals of $s_1 = \{Goal, 1-0\}$ and $s_2 = \{Goal, Ramires\}$ have quite similar PSD

Algorithm 5. The main algorithm for extracting the events' characteristic sets

Data: KW - The set of keyword signals;
γ - The cut-off value for determining the cluster's members, $\gamma \in [0:1]$;
Result: CW - List of characteristic sets

$CW = \emptyset$;
for $\forall w1 \in KW$ **do**
  $KW = KW \setminus \{w1\}$;
  $w2 = \arg \max_{\forall w \in KW} ccl(PSD_{w1}, PSD_w)$;
  $C = \{w1, w2\}$;
  for $\forall w3 \in KW,\ w1 \neq w2$ **do**
    if $ccl(PSD_{w1,w2}, PSD_{w1,w3}) > \gamma$ **then**
      $C = C \cup \{w3\}$;
      $KW = KW \setminus \{w3\}$;
  $CW = CW \cup \{C\}$ if $|C| > 2$

pattern and intensity, the PSD of $s_3 = \{Goal, Drogba\}$ follows another style. In this case the similarity score using normalized cross-correlation function ccl without time delay between pairs (s_1, s_2), (s_1, s_3), (s_2, s_3) are 0.85, 0.57 and 0.6 respectively. This property gives us an important distance measurement to determine the cluster of each certain keywords as showing in Algorithm 5.

4 Experimental Results

We have built an application with two separate modules and they cooperate each other for collecting and analyzing data from Twitter [5]. The implementation of proposed method works well on an input dataset containing a set of strict content-related tweets rather than a group of tweets with fragmented content. For evaluating the efficiency of the proposed method we compare our detected event list on a tweet dataset collected around The Final football match between Chelsea and Liverpool in FA Cup 2012. The database is introduced by Aiello et al. in [2]. It is used for evaluating performance of several classic event/topic detection methods. Authors also release an evaluation tool for comparing the event detection result between these methods, so that it is equivalent for comparing our proposed method with the introduced ones.

Table 1 Comparison of event/topic detection based on ground truth topics. T-REC, K-PREC, K-REC refers to Topic Recall, Keyword Precision and Keyword Recall respectively

Dataset	**FACup**		
Method	T-REC	K-PREC	K-REC
BNgram	**0.769**	**0.355**	**0.587**
Proposed method	0.692	0.315	0.533
LDA	0.692	0.230	0.511
Doc-P	0.615	0.311	0.559

With $\Delta T = 1$ minute and $\gamma = 0.85$, our proposed method gives out a promising result ranked at the second position. However, the keyword precision and recall are small as showing in Table 1. The reasons may be that we only analyze keyword signals independently but the combination of these keywords in microposts is not really considered. While we know that the events/topics are strongly depended on the context of data; also the SNS network structure and the information propagation speed are significant features.

5 Conclusion

In this work, we demonstrated an approach to analyze social data in a new dimensional space but giving an equivalent efficiency as same as processing the data in its original format. Its result is quite promising, but it also point that besides the sudden variation in signal of keywords we need to consider other significant features such as: the co-occurrence of keywords in the microposts; the structure of SNS network; the speed of information propagation and so on. In the future work we will continue solve the issue by proposing an approach which integrates these features consistently.

Acknowledgements. This work was supported under the framework of international cooperation program managed by National Research Foundation of Korea (NRF-2013K2A1A2055213). Also, this work is supported by BK21+ of National Research Foundation of Korea.

References

1. Aggarwal, C.C., Subbian, K.: Event detection in social streams. In: Proceedings of the 12th SIAM International Conference on Data Mining, Anaheim, California, USA, April 26-28, pp. 624–635 (2012)
2. Aiello, L.M., Petkos, G., Martín, C.J., Corney, D., Papadopoulos, S., Skraba, R., Göker, A., Kompatsiaris, I., Jaimes, A.: Sensing trending topics in twitter. IEEE Transactions on Multimedia 15(6), 1268–1282 (2013)
3. He, Q., Chang, K., Lim, E.P.: Analyzing feature trajectories for event detection. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2007, pp. 207–214. ACM (2007)
4. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)
5. Nguyen, D.T., Jung, J.E.: Privacy-preserving discovery of topic-based events from social sensor signals: An experimental study on twitter. The Scientific World Journal 2014, Article ID 204785 (2014)
6. Weng, J., Lee, B.S.: Event detection in twitter. In: Proceedings of the Fifth International Conference on Weblogs and Social Media, ICWSM 2011, Barcelona, Catalonia, Spain. The AAAI Press (2011)
7. Zhou, X., Chen, L.: Event detection over twitter social media streams. VLDB Journal 23(3), 381–400 (2013)

Modeling Open Accessibility Data of Public Transport

Paloma Cáceres, Almudena Sierra-Alonso, Carlos E. Cuesta,
Belén Vela, and José María Cavero

Abstract. Nowadays linked open data (LOD) is a great challenge in the area of information technologies. In fact, some public administrations and organizations are working to open their data to citizens. LOD describes a method of publishing structured data to become more useful. Heterogeneous distributed data sources which have been published as LOD, could be integrated. The information of public transport networks is of public interest. A classic example is the route planning systems which combine data from different public transit networks. Besides, a relevant aspect of the public transit networks is the accessibility of this media which make possible the mobility of special needs people. Due to the large amount of these data, LOD provides the mechanism to publish them and to support new mobility services to the citizens. The aim of this work is present the process to define public transit data and their accessibility information as LOD from diverse data sources taking into account some of the main reference data models for public transport: IFOPT.

1 Introduction

Nowadays the Linked Open Data (LOD) [1] initiative emerges as a great challenge in the area of information technologies. It consists of providing the mechanisms for publishing, enriching and sharing data, information and knowledge on the Web, using semantic web technologies. LOD is based on the following principles: (a) using Universal Resource Identifiers (URIs) [2] to identify all kinds of "things", (b) making these URIs accessible via the HTTP protocol and (c) providing a description of these elements using the Resource Description Format (RDF) [3], along with (d) URI-based links (again) to related information.

Paloma Cáceres · Almudena Sierra-Alonso · Carlos E. Cuesta · Belén Vela · José María Cavero
VorTIC3 Research, Dept. of Informatics & Statistics,
School of Computer Science & Engineering (ETSII),
Rey Juan Carlos University, Madrid, Spain
e-mail: {paloma.caceres,almudena.sierra,carlos.cuesta,
belen.vela,josemaria.cavero}@urjc.es

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_25

This approach can be applied to multiple heterogeneous distributed data sources, which might be integrated and then published as LOD. In particular, a specific application could be the management of information related to public transport networks, which is of public interest. Their amount and size makes difficult to share these data: LOD provides the mechanism to publish them, and to support the definition of new mobility services for citizens. A classic example might be route planning systems, which combine data from different public transit networks. Even within this specific context, we can find particularly relevant aspects. We are particularly interested in the question of accessibility: specifically, how to tackle the problem of exploiting information about public transit networks for special needs people. Specifically, we would like to explore this challenge, namely how to provide accessibility information for a user within a transit planning system, in *real time*.

This problem is actually a real-world-scale concurrency problem; in particular consider the case in which the planning system returns a route which implies some transfers (between different transport media). The approach is implicitly parallel: as the route is composed of different subroutes, we can obtain the transport information for each route separately (i.e. simultaneously), including in particular accessibility data for this subroute. At the same time, transfers are themselves points of contact – and then every transfer defines a concurrency problem. Our goal is to obtain the accessibility information for every composite route (using LOD as already explained), as queried by a specific user, in real time, and including the synchronization between different transport media with regard to every transfer. The Spanish National Society of Blind People (ONCE) [4] has expressed their interest in the results of this work.

To the best of our knowledge, the accessibility information of the public transport network is not easily available as open data. Our proposal focuses on the public bus network of Madrid, and their accessibility features. This information has been provided by EMT Madrid [5] which is the public bus company in Madrid. We have defined a process where we analyze the original data, and then we identify the data semantics about accessibility information. Later, we contextualize these data using the vocabulary of the transport metamodel IFOPT [6]. We chose IFOPT because it incorporates specific structures to describe accessibility data about the equipment of vehicles, stop places and access areas. Finally, we have modeled them in RDF to be able to expose them as linked open data.

This work is a part of the CoMobility project [7], which proposes an IT platform, to assist in intermodal transport sharing, and to integrate the use of carpooling with public transport, as well as private transport media.

The paper is structured as follows: in section 2, we briefly introduce the context of this work, the CoMobility project; section 3 describes our proposal which defines a process to publish accessibility data of Public Transport in LOD, including an example; finally, the main conclusions are shown in section 4.

2 Context: The CoMobility Project

The CoMobility project defines a multimodal architecture based on linked open data for a sustainable mobility. Its main goals are improving the citizen mobility, optimizing their trips combining both public transport and private sharing transport (i.e. car sharing), providing accessible trips when necessary and saving energy and reducing the pollution.

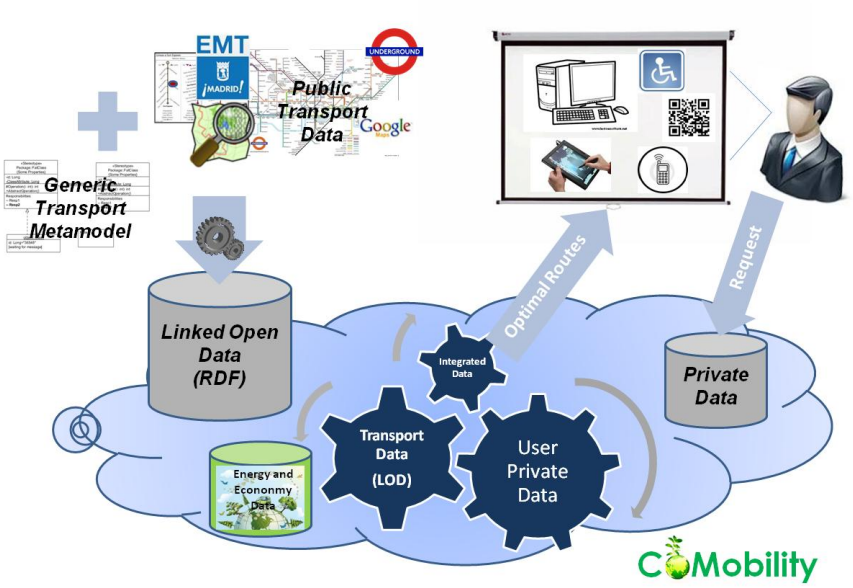


Fig. 1 The CoMobility Project structure

We have developed a systematic approach to (i) accessing open, integrated and semantically annotated transportation data and street maps, (ii) combining them with private data, and (iii) supplying mechanisms to allow the actors to share and search these data. Therefore, its conceptual architecture provides the means to perform the following tasks. First, the platform can identify, select, extract and integrate data from different and heterogeneous sources, stemming from the transportation, geographical and energy domains. Second, data from public institutions is obtained automatically in the form of open data. Third, these data are annotated as linked data, and a set of heuristics generate links between data items from different sources without human intervention. Fourth, these data are integrated with private data provided by users themselves. And finally, CoMobility provides intuitive and customized data analytics and visualization, allowing individuals to become aware of the environmental impact of their transport choices. Figure 1 provides a general idea about our project.

3 Modeling Accessibility Data of Madrid Public Bus Network in Linked Open Data

This work focuses on modeling accessibility data for the public bus network using Linked Open Data (LOD). We would like to emphasize two features of this work. First, that we have worked with real accessibility information and data from the Madrid public bus network, provided by EMT Madrid (the public company). Second, that in its conception, we have followed the IFOPT metamodel, the current standard to model the features of transport media, including accessibility. IFOPT defines a model and the identification principles for the main fixed objects related to the access to Public Transport (e.g. stop points, stations, stop areas, connection links, entrances, etc.). In particular, our work in this paper focuses specifically on a subset of the *Stop Places* model, namely the *Vehicle Equipment* submodel. For the remainder of this paper, we will focus in the vehicle equipment which models accessibility features.

To achieve this goal, we have defined a stepwise refinement process, which we briefly describe in the following:

- First, we have studied the accessibility features of the public bus network, and the format in which the original data is provided. Later, we have identified the data semantics of our accessibility information;
- Second, we have analyzed the accessibility information included in the original IFOPT metamodel, and we have matched the original data of EMT Madrid against the vocabulary of this metamodel;
- Third, using the metamodel as a basis, we have defined an ontology that describes the accessibility features of the public bus network, specifically in the domain of vehicle equipment;
- Then, considering both this ontology and the available information, we have semi-automatically generated a RDF Schema [8] for this ontology;
- After this schema is completed, we have distilled the set of RDF data, by instantiating the concepts in our RDF Schema with real data about specific vehicles from the EMT Madrid network;
- Finally we have published this RDF dataset as Open Data, in a format which can be retrieved using SPARQL [9].

Fig. 2 shows the portion of the CoMobility project in which the current work is focused, specifically the part that involves integration of PT data in LOD format.

In the next subsection, we detail the process by means of an example. We describe the process to be performed to publish the accessibility data, available for specific vehicles from EMT Madrid, with regard to the information about wheelchairs.

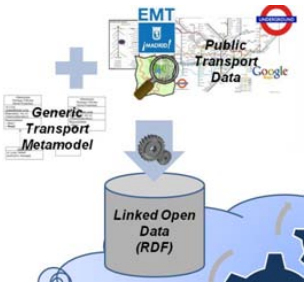


Fig. 2 The focus of our proposal – transport data in LOD

3.1 Case Study: Providing Wheelchair Accessibility Information within Madrid Public Bus Network

As already noted, the goal of this work is to make the accessibility data for the public bus network available as open data, matching the IFOPT standard metamodel. We have followed the process described in the previous section to transform these data from the original information into an open semantic format.

We have studied the accessibility features for the public bus network [5,10], which we have obtained in a specific, private format, as provided by EMT Madrid. We have analyzed the data semantics for this accessibility information. In this particular case study, we have found that some vehicles have a specific area for wheelchairs, in which both the height and the width of access area, and the turning circle area (to turn the wheelchair) are also specified. Then we have compared this information to the accessibility elements in the IFOPT metamodel. Specifically, IFOPT provides a class for this purpose, namely the *WheelChairVehicleEquipment* class, in its UML version, which specifies the following attributes:

- *NumberOfWheelChairAreas*[0..1]:*integer*
- *WidthOfAccessArea*[0..1]:*metres*
- *HeightOfAccessArea*[0..1]:*integer*
- *WheelChairTurnningCircle*[0..1]:*metres*

As we can see, IFOPT supports all the required information about wheelchairs, for the actual network of public buses from EMT Madrid. Then considering this information, we have defined an ontology for it, describing the domain of vehicle equipment, and specifically focusing on the accessibility features for public buses.

This ontology summarizes the knowledge contained in the UML version of the IFOPT submodel *Vehicle Equipment*, as already indicated. The structure of this class diagram has been described directly into Apache Jena, accessed from Eclipse [11]. Apache Jena [12] is a free and open source Java framework for building Semantic Web and Linked Data applications.

Afterwards, we generate the RDF Schema [8] for the defined ontology, using a semi-automatic process which involves both Jena [12] and Protegé [13]. Fig. 3 shows a part of that RDF schema, dealing with the vehicle equipment. We can see how the

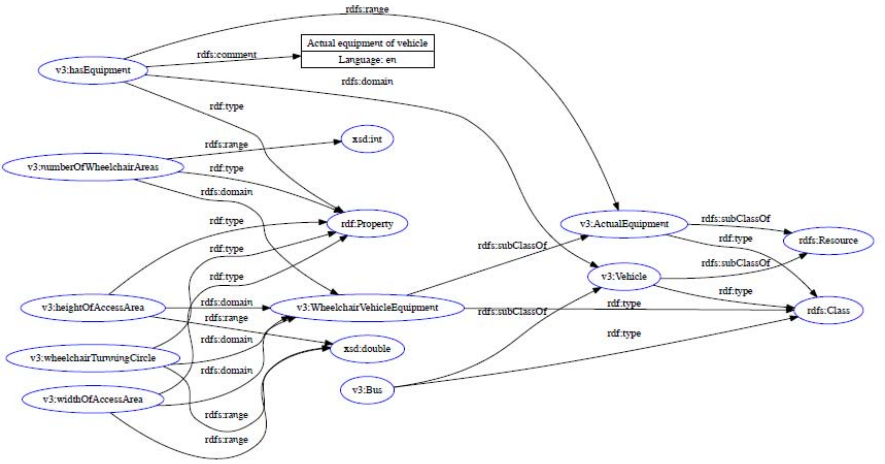


Fig. 3 Fragment of the RDF Schema for wheelchair data

information of the UML class diagram is maintained. In the UML class diagram, the `WheelchairVehicleEquipment` class has four attributes, which are modeled in the RDF Scheme with rdf:domain predicates. Furthermore, it is also indicated that this class is a subclass of the class `ActualVehicleEquipment` by a connection defined through the predicate *rdf:subClassOf* with the subject `v3:ActualEquipment`.

For the last step of the process, we have generated the RDF dataset for this information, using the real data provided by EMT. According to these data, a specific bus kind has two Wheelchair Areas. Each one of them has a width of access area of 1.6 metres, a height of access area equal to 2.6 metres and the turning circle area is 3.15 metres. The RDF data diagram for this information is provided as Fig. 4.

As we can see, the specific set of RDF data tells us that the bus *Bus_1* has the equipment for two wheelchairs (predicate `v3:numberOfWheelchairAreas`)

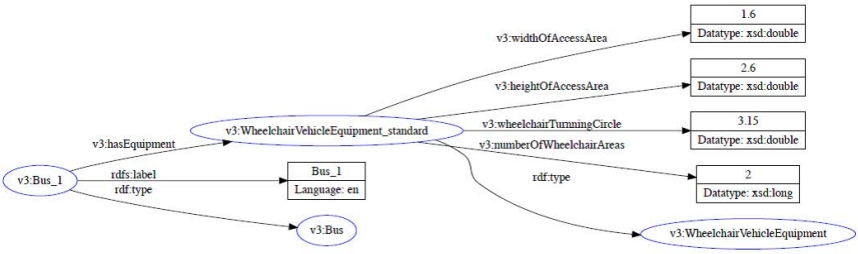


Fig. 4 RDF data diagram for the wheelchair equipment in the *Bus_1*

and defines the metrics of the access area (predicates `v3:heightOfAccessArea`, `v3:widthOfAccessArea`; and also `v3:wheelchairTurningCircle`).

Once obtained and fully populated, this model has been checked against the previous RDF Schema, using again Jena for that purpose. Therefore, the RDF dataset has been shown to be fully compliant to that RDFS.

After this verification, we are finally able to publish the full dataset, with this accessibility information. We could use any RDF store to make this available online; but for convenience, we use Fuseki [12] (as it is, in turn, another element of the Apache Jena Project) to publish the data. Fuseki already provides the required capabilities to be able to query this information, using SPARQL [9], therefore making it available as open data. We also provide a Jena-based programming interface to access the same information, mostly to perform internal queries and directly manage the data.

4 Conclusions

This paper presents the process to publish the data of public transport networks as Linked Open Data. Our purpose is to be able to offer a new kind of services (such as route planning) for public transport users, particularly for those with special accessibility needs. To the best of our knowledge, the accessibility information of the public transport network is not easily available as open data.

In this case, the new service that we propose consists of determining the accessibility features for an intended route, designed by a planner. The route is the solution for a certain query by a specific user; it is composed by different subroutes and transfers, which are computed in parallel and must resolve concurrency issues in real time.

We propose a process to transform the accessibility information of the public bus network of Madrid into Linked Open Data. We have studied data accessibility from the original source (EMT Madrid) and the IFOPT metamodel. Later we have matched the data with the metamodel, to determine which accessibility elements are included or not. Next, we have defined the ontology that describes the specific accessibility vocabulary, to take it into account. Then we have distilled the corresponding RDF Schema, to finally generate the information in RDF, according to this schema, including the specific accessibility data of a bus, to validate the final process. We have used Apache Jena, among some other platforms, in this process.

We conclude that it is possible to publish accessibility data of public transport network in Linked Open Data.

Acknowledgements. This work has been supported by the project CoMobility (TIN2012-31104), funded by the Spanish Ministry of Economy and Competitiveness; and it has also been supported by the Chair of Ecotransport, Technology and Mobility (`http://www.catedraetm.es/`) at Rey Juan Carlos University.

References

1. Linked Open Data, http://linkeddata.org/
2. URI Planning Interest Group. URIs, URLs, and URNs: Clarifications and Recommendations 1.0 (September 2001), http://www.w3.org/TR/uri-clarification/
3. Klyne, G., Carroll, J.: Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation, http://www.w3.org/RDF/
4. Spanish National Society of Blind People, http://www.once.es/new
5. EMT Madrid home page (2012), http://www.emtmadrid.es/
6. IFOPT: Identification of Fixed Objects in Public Transport. prCEN Technical Standard (March 2013), http://www.dft.gov.uk/naptan/ifopt
7. Cuesta, C.E., Cáceres, P., Vela, B., Cavero, J.M.: CoMobility: A Mobile Platform for Transport Sharing. ERCIM News 93 (2013)
8. RDFS: Resource Description Framework Schema (February 2004), http://www.w3.org/2001/sw/wiki/RDFS
9. SPARQL Query Language for RDF (January 2008), http://www.w3.org/TR/rdf-sparql-query/
10. Consorcio Regional de Transportes de Madrid (CRTM). Accessibility on Public Transport in Madrid. Ed. Comunidad de Madrid (2013)
11. Eclipse, http://eclipse.org
12. Apache Jena, http://jena.apache.org
13. Protegé, http://protege.stanford.edu

Part VIII
Machine Learning

A Machine Learning Attack against the Civil Rights CAPTCHA

Carlos Javier Hernández-Castro, David F. Barrero, and María D. R-Moreno

Abstract. Human Interactive Proofs (HIPs) are a basic security measure on the Internet to avoid several types of automatic attacks. Recently, a new HIP has been designed to increase security: the Civil Rights CAPTCHA. It employs the empathy capacity of humans to further strengthen the security of a well known OCR CAPTCHA, Securimage. In this paper, we analyse it from a security perspective, pointing out its design flaws. Then, we create a successful side-channel attack, leveraging some well-known machine learning algorithms.

1 Introduction

The abuse of free web-offered services was already a big issue in the 90's. First approaches to solve the problem proposed theoretical methods to prevent such attacks [10], under the idea of using problems (thought to be) hard for computers, but easy for humans, that is, Human Interactive Proofs (HIPs). Some years after, CMU Researchers improved this idea with a program to tell bots from humans apart, and listed the desirable properties such a program should have. Thus, the term CAPTCHA was coined. Their program used english words chosen at random and rendered them as images of printed text under a wide variety of shape deformations and image distortions. The user was asked to transcribe some minimum number of any of those words correctly.

During the 2000s decade they was a lot of research on new techniques [6, 9, 16] enabling the breaking of text-based word-image CAPTCHAs. All of these attacks

Carlos Javier Hernández-Castro
Universidad Complutense, Madrid, Spain
e-mail: chernandez@ucm.es

David F. Barrero · María D. R-Moreno
Computer Engineering Department, Universidad de Alcalá, Madrid, Spain
e-mail: {david,mdolores}@aut.uah.es

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_26

were not really improvements of the state of the art in computer optical character recognition (OCR). Instead, they made a clever use of some very simple properties of the challenge images to undo part of the distortions or extract enough information from them [3]. Added to some design flaws, this allowed attackers to *read* them. Some companies created too difficult HIPs [5], whereas many researchers started looking into the broader AI problem of vision and image analysis.

Image-based CAPTCHAs need a large-enough database of labelled pictures. Ahn and Dabbish proposed a new way by creating a game, the "ESP game". The site Hot-Captcha.com was the first to propose using a large-scale, human-labeled database. Oli Warner proposed using photos of kittens to tell computers and humans apart [14]. Another proposal, the HumanAuth CAPTCHA, asks the user to distinguish pictures depicting either a nature-related image (e.g. a flower, grass, the sea), or a human-generated one (e.g a clock, a boat, or Big Ben). ASIRRA uses a similar approach based on cat/dog classification of images, but uses a large database "of more than 3 million photos from `Petfinder.com`". All these proposals have also been broken. Other, typically based on image analysis [13], like face classification [4], or even cartoons [15] had appeared recently, and await scrutiny. There are also some new proposals enhancing the typical OCR/text-based HIP [1, 8].

In this article we present a novel attack against the Civil Rights CAPTCHA, an original CAPTCHA that aims to join the strength of a typical word-recognition CAPTCHA, reinforcing it with an all-new empathy challenge. This combination purposely leads to a stronger, more secure CAPTCHA overall, and at the same time, makes users aware of Civil Rights news around the World.

Our attack is a side-channel attack, as it does not try to solve neither of the Artificial Intelligence (AI) problems used as a foundation for the CAPTCHA: it does not solve the empathy problem, nor the word recognition problem, as general AI problems. Instead, it solves both of them for this particular instance, or problem subset. We achieve so by identifying the security problems in the design of this HIP, and applying well-known Machine Learning algorithms to exploit them.

In the following sections, we first introduce the Civil Rights CAPTCHA (*CRC* from now on). We discuss the different design flaws found while closely examining it (section 3), and present a novel attack exploiting them. This attack uses some well-known machine learning algorithms to automatically solve it (section 4). Section 4 shows the results obtained using this approach. Finally, conclusions are outlined.

2 Civil Rights CAPTCHA

The *CRC* is based on the human ability to show empathy after being presented with a news excerpt, typically containing some news about Human Rights and/or Civil Rights around the World.

This CAPTCHA is based on Securimage, a word-distortion CAPTCHA (Fig. 2). How this Civil Rights CAPTCHA is supposed to work is simple: *CRC* picks up a Civil Right news from its DDBB, and uses Securimage to create three possible

answers, later presented to the user as distorted images containing words describing feelings (i.e. "agitated", "happy" and "angry"), who should write down the correct one based on the emotions originated from the news headline presented to her. This news bit is related to Human or Civil Rights, and supposed to create an emotion out of the empathy of a human reader. As a result, if we consider the CRC well designed, and Securimage security provides a security level *X*, this CAPTCHA design should provide an increased $3 \times X$, because picking up the correct answer should not be easier than random guessing. As we will see in brief, the security of this CAPTCHA is unfortunately below that *X* security level of Securimage.

CRC is provided as a service, directly accessible using an API allowing a programmer to connect to it, download a challenge composed of a news text, and three images containing one or two words distorted using *Securimage*, one of them the correct solution to the challenge. The same API allows to send to the *CRC* server the text the user inputs, to check if this is a correct answer (human) or not (program).

In order to analyze the *CRC*, we decided to analyze its client-server communication from the end-point, the same viewpoint a real attacker would have. We used a HTTP traffic analyzer . After a few interactions and tests, we were able to decipher the core of the client-server protocol needed in order to program an attack:

1. The first step is to request the main content for the CAPTCHA form, located at `http://captcha.civilrightsdefenders.org/captchaAPI/`. This, if presented as-is to the user, will be an *empty* HTML structure where the real content (news-bit, answer images) will be plugged in later using JavaScript. Another important function of this HTTP answer is to set the value of the *ci_session* cookie. This cookie is a meta-cookie containing several bits of information, like a *session_id*, the *IP* address of the client, information about its *user-agent*, etc.
2. Once this is loaded, the client JavaScript code makes another request with the same URL and *?sessid=1*. This sends back an answer containing the *PHPSESSID* cookie. This is the one the PHP library uses to keep track of client sessions.
3. In the next request, the parameters *callback, newtext* and *lang* are added to the URL, causing the server to send back the text of the challenge - the news bit. This comes back as a JSON encoded text.
4. The three images containing the answer are downloaded, each one requesting an unique (and random) 20-character id. The server keeps track of the ones to send using the previously provided cookies.
5. After the user has written the answer, this is sent to the server encoded in the URL of the next request.

3 Civil Rights CAPTCHA Design Flaws

After the analysis of the protocol, the first question raised naturally: what would happen if we keep asking the server for more word-image answers. We did, and confirmed that the server keeps providing us with new word-images, independently of adding or not the *newset* parameter.

This again raised a question: would it be possible that the server does not keep track of the word-images sent, and only checks that the answer is valid (positive, negative) according to the news bit?

We checked this hypothesis. In particular, we wrote down a few positive answers and a few negative ones from another questions. Then we proceeded to the next question, *"In October 2012 the Ukrainian parliament took the step to approve a law, which criminalises 'propaganda of homosexuality'. How does that make you feel?"*. The corresponding word-image answers were *very crappy, elastic* and *hopeful*. Being a negative news bit, we decided to answer with a negative answer present in other questions but not in this one, choosing *horrified*. Result: error! We tried the same attack a few more times, without suceess.

Our conclusion at this point was that either the answers are divided in finer categories or, more probable, the server keeps track of the word-image answers sent (probably the last three). To find out the correct hypothesis, we proceed with the attack (section 4), collecting logs of wrong and correct answers. Then, we tried again using these logs to find correct answers for each question. Again, we got a fail. The only possible conclusion is that the server keeps track of the word-image answers sent.

Once we finished playing around with the *CRC* and got familiar with how it operates, we wanted to analyze the challenges it could present to the users. We wanted to learn their number, distribution, and if any characteristic of the was not uniform. For this purpose, we wrote a program able to follow the protocol of the *CRC* (gathering and sending back the cookies, etc.), mimicking a regular user, download and interpret the information.

After downloading 1000 challenges, we learned that there are only 21 different challenge texts. This is in itself a flaw, as it is too low for a CAPTCHA.

We also checked how many times each challenge has been presented during this test. The questions have a seemingly uniform distribution of appearances, with a χ^2_{20} with a p-value of 0.336 (Fig. 1).

Each challenge comes with three different answers. How many total answers are there? During our experiments, we have been able to observe 130 different answers, 28 of them compositions of the words *quite, really, truly* and *very* and some of the remaining 102 basic categories. Again, this is a problem. A CAPTCHA with only

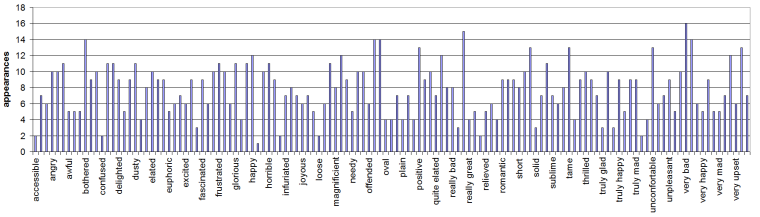


Fig. 1 Appearances of each question

130 possible answers is a CAPTCHA that can be broken 0.76% of the time just by answering any of them, already a bad result for any CAPTCHA.

The answer type distribution is not uniform:, there are 54% answers describing a negative emotion, and 40% describing a positive one (plus 6% describing no emotion, like the answers *accessible, oval, plain, temporary, typical...*). The distribution of their appearance (from the 1000 manually classified ones) seems to be uniform within the different categories, with $58,3\%$, $38,3\%$ and $3,4\%$ for each corresponding category. Surprisingly, it does not seem to be *uniform*, as the p-value of its χ^2_{129} is 0.00365. This can be further exploited for a blind, brute-force attack, although this is not our purpose here, as our attack will produce better results.

4 A Machine Learning Approach to Attack the Civil Rights CAPTCHA

In this section we will introduce our attack, that can be divided in *reading* the Securimage-protected answers, and later, classifying the challenge text. Given the design flaws of the *CRC*, classifying the challenge text is not strictly necessary, but does improve the results. We specify how we use Machine Learning for solving both problems to a level that breaks the *CRC*.

4.1 Classifying the Answers

The current iteration of Securimage might be a good OCR-based CAPTCHA, but the way it is employed in *CRC* makes it quite weak. The problem is Securimage was originally designed to work with a large alphabet, and either random words, or a huge dictionary. If we restrict it to just 130 words, its disguising capabilities might not be good enough for a strong classifier. This is what we decided to test.

Probably an in-depth analysis of the Securimage-produced images would be able to break it more accurately, in this case of only 130 possible answers. We did not want to produce such an attack, but to check if a very basic analysis of the same images, and the use of Machine Learning, would be able to break it.

The statistics we gathered from the images, to feed our AI-classifier, were typical ones: black pixel count, and pixel count per column. This, though, will be affected by the presence of the two black lines (see figure 2).

The two lines drawn are typically of the same thickness all along the length of each one. In this case, a derivative of the number of the vertical pixels in each column will be affected by their presence only at their start and their end (ideally, as intersection of lines and letters will affect too). We also added this statistic.

We decided to sample the image at every column, and then in groups of 3, or 5 columns; and to use a very simple approach - our intent is to do the least possible analysis and let the machine learning algorithm do it for us. Kind of the approach

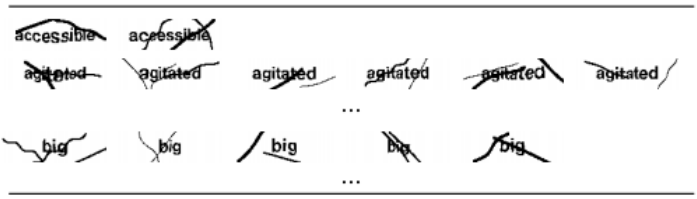


Fig. 2 CRC-Securimage answers

we would expect from a *lazy* hacker wanting to obtain results as fast as possible. So we calculated all the statistics, for each image, and let the different machine learning algorithms cope with the data.

To provide some knowledge for training, we downloaded 1000 answer word-images , and manually classified them into the 130 possible categories.

After feeding them to Weka [7], and using its different classifiers, always using 10-fold CV, we realized that best out-of-box classification was obtained with J48 trees [12]. An astonishing 26.7% of the time we were able to correctly *read* the answer.

4.2 Classifying the Challenge Text

Given the *CRC* design flaws, it is possible to do better. *CRC* presents only 21 different questions, 6 of them positive (29%), and 15 negative (71%). The problem with this is that a *lazy* all-negative classifier will have a 71% success ratio, so we can improve the success of our attack by discarding all read answers that are positive.

This result can be improved. Several projects are available to classify the emotion of a written text [2,11]. The problem with most of them is that they typically classify the emotion of the writer who wrote it by checking the adjectives and/or nouns used. This is not suitable to our case, because the news can be objectively described (no or little use of adjectives), but still be able to create an empathy emotion on the reader (according to the positive or negative impact of it on other people). For our case, one possible approach would be to use the Python *N*atural *L*anguage *T*ool-*K*it (*NTLK*) library [2]. This library provides several algorithms for treating Natural Language problems, some of them to classify text, including decision trees, maximum entropy, SVMs or Naïve Bayes, just to mention some. Among these possibilities, we chose the Naïve Bayes algorithm, which we feel will suit most an attacker looking for a low-cost breach.

To be able to train our model we needed a set of news-bits to be manually classified as either positive or negative. We found two main sources, the Human Rights Watch (HRW) association (`http://www.hrw.org/news`), and the Civil Rights Defenders (CRD) (`http://www.civilrightsdefenders.org/category/news/`).

These last ones happen to be the ones associated with the *CRC*. We downloaded 152 news from the Human Rights Watch association (most of them of negative content), and 622 from the Civil Rights Defenders.

During the set-up of our Bayes classifiers, we were careful to take out of the bags of words the name of any country (and corresponding adjectives), name of the civil & human rights organisations and related organisations, and of course, the NLTK stopwords for English, so they are not used for classification.

Training our model with the HRW association news, we were not able to attain better success in news-type classification. This can be because, from the 152 news excerpts, only 5 are of positive content. Training it with the CRD, we could obtain a 76% success ratio, by improving the classification by an additional correct detection.

As there are only 21 challenges, we can memorize their positive/negative classification. The reason why we do not do it here, and instead try this ML approach, is so we can successfully answer the question "if the number of challenges is raised properly, and actively maintained, could we still successfully attack this CAPTCHA?".

Still there is another, slower but more precise way of classification. Imagine we are presented with a new challenge, and we, as an algorithm, do not know whether the correct answer should be positive, or negative. We can still *read* each answer 27% of the time, and being a low number of them, classify them as either positive or negative. Thus we can keep answering randomly from the answers we read, and when we succeed, by looking at the type of answer that was successful (positive/negative), we can correctly classify the new challenge text.

In brief: adding this questions does not seem to have a huge beneficial impact of the security of this CAPTCHA. The fact that the empathy classification of these questions seems to be quite coarse (positive/negative) means it will not significatively add security to the CAPTCHA.

5 Experimental Results

In this section we explain in detail the implementation and results of our attack. This attack was slightly improved during its development, in what we introduced here as the *improved attack*. We compare the results of both. These two attacks join the results of the previous sections, showing that the use of Machine Learning to exploit the *CRC* design flaws is indeed able to break it.

5.1 Basic Attack

The attack was coded using the Python libraries *urllib, urllib2, cookielib, json* and *nltk*, and also calling Weka for classification of the answer images (*reading* them). While doing so, we saved a log with all the relevant information, for further exam.

Most importantly, we kept track of the answers returned by the *CRC* server as correct ones - passing the CAPTCHA as humans.

The basic attack is quite simple: after downloading the challenge text and images, it classifies the text (neg./pos.) according to our model, classifies the answers using the J48 tree, removes those answers with a sense (pos./neg.) not according to the question. Among the remaining answers, the J48 tree gives to us not only their classification but also a 0-1 index of security of that classification. So the attack chooses randomly one, giving more weight to more securely *read* ones.

5.2 *Improved Attack*

While designing and testing our basic attack, an improvement was tried out. The idea is simple: to remove answers that we know that are not correct for each question (by previous failures). Not only that, if among the answers is present one that we know is correct, then use it. We modified the logic of our attack to accommodate these new ideas, and checked how well it improved our results over time (that is, as our knowledge base of errors and successes was growing).

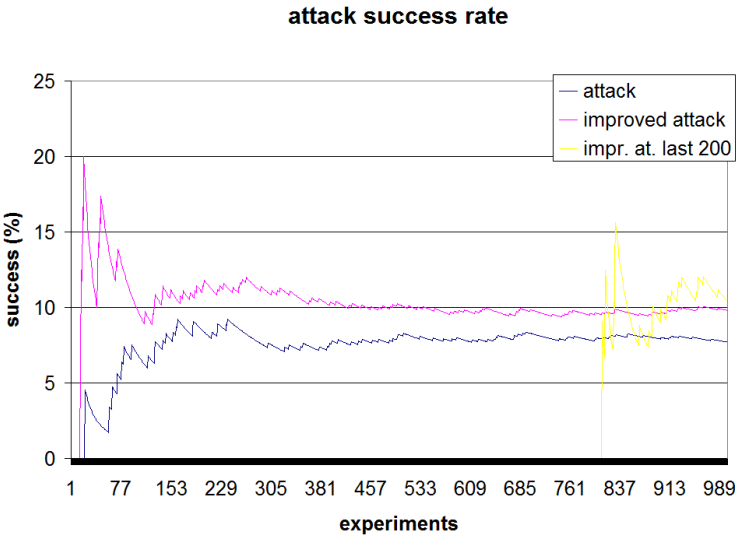


Fig. 3 Percentage of success rate of the attacks

5.3 Results

Due to technical problems, including the large amount of time taken by the *CRC* server to provide a full challenge (45 secs. during our tests, including the three corresponding images), and the low reliability of the same (since the server stopped responding for some minutes on a few occasions), it was easy to incur into timeouts, and difficult to finish a large series of experiments.

With these restrictions, our lengthiest experiments consists of series of merely 1000 challenges for the basic attack (took 17 hours and 54 minutes), and the same for the improved version of the attack (took 14 hours and 55 minutes).

In figure 3 you can see the ratio of success of both attacks as they evolve[1]. The first one is clearly stable at 7.7%, whereas the ratio of the improved version augments with the gathered *knowledge*, with a mean for the total experiment of 9.8%.

If we focus on the last 200 challenges presented to the improved version of our attack (when the *knowledge* has gained some size), figure 3 shows that the success ratio of breaking the *CRC* is 10.5%.

This result is better than a brute-force attack, that would break the *CRC* on average $\frac{1}{130}$ $(0,77\%)$, or $0,71 \times \frac{1}{70}$ $(1,01\%)$ if we restrict ourselves to negative answers. A brute-force attack would not be able to learn the correct answers to each question, as it is not *reading* which answers are present each time. If somehow an attacker creates a DDBB of correct answers to each question, and then uses it to answer a random correct answer, its success ratio would never be over $\frac{\sum_{i=1}^{i=21} \frac{1}{|solutions(i)|}}{21}$, where $solutions(i)$ is the set of all possible correct solutions to question i[2]. In an scenario of a well maintained DDBB of challenges, such an attacker would take extremely long to *learn* all the right answers to all the questions. Our attack will still be able to attain a minimum 7.7% success ratio. Once automatically learned some correct and wrong answers, a 10.5% success ratio or greater would be possible.

6 Conclusions

In this article, we analyze the Civil Rights CAPTCHA from a security standpoint, using Machine Learning algorithms to consistently break it at 10.5% success ratio.

We show how a CAPTCHA, Securimage, is rendered useless by using it out of the scope it was designed for. We also show that the idea of a CAPTCHA based on empathy about other subjects is not necessarily good, especially if this empathy test can only be administered as a choice between two main categories (or a small number of categories). Finally, we have shown that the combination of two CAPTCHAs is

[1] Calculated as $\frac{sucess}{total}$, so it is more prone to variation with a lower number of experiments.

[2] This would be in a scenario in which the attacker has learned *all* possible right answers to each question. Even in our 1000-length attacks we were not able to learn all the correct answers, with some questions having 5, 7 or 9 correct answers, but others still none.

not always more secure than one of them alone, as the way *Securimage* is used by the *CRC* lowers its security, and in turn allows us to break the *CRC*.

We plan to further improve on this attack by analyzing different statistics and ML algorithms against both the *CRC* and *Securimage*. We also plan on using the same security analysis techniques and ideas to check the security of other HIPs.

Acknowledgements. The first author wants to thank Zhenya for her Mashka i Dashka, i neë krasivaya ulybka.

References

1. Alsuhibany, S.A.: Optimising captcha generation. In: 2011 Sixth International Conference on Availability, Reliability and Security (ARES), pp. 740–745 (August 2011)
2. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly, Beijing (2009)
3. Bursztein, E., Martin, M., Mitchell, J.: Text-based captcha strengths and weaknesses. In: Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS 2011, pp. 125–138. ACM, New York (2011)
4. D'Souza, D., Polina, P.C., Yampolskiy, R.V.: Avatar captcha: Telling computers and humans apart via face classification. In: 2012 IEEE International Conference on Electro/Information Technology (EIT), pp. 1–6 (May 2012)
5. Fidas, C.A., Voyiatzis, A.G., Avouris, N.M.: On the necessity of user-friendly captcha. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2011, pp. 2623–2626. ACM, New York (2011)
6. Golle, P.: Machine learning attacks against the asirra captcha. In: Proceedings of the 5th Symposium on Usable Privacy and Security, SOUPS 2009, Mountain View, California, USA, July 15-17. ACM International Conference Proceeding Series. ACM (2009)
7. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: An update (2009)
8. Kouritzin, M.A., Newton, F., Wu, B.: On random field completely automated public turing test to tell computers and humans apart generation. IEEE Transactions on Image Processing 22(4), 1656–1666 (2013)
9. Mohamed, M., Sachdeva, N., Georgescu, M., Gao, S., Saxena, N., Zhang, C., Kumaraguru, P., van Oorschot, P.C., Chen, W.B.: Three-way dissection of a game-captcha: Automated attacks, relay attacks, and usability. CoRR, abs/1310.1540 (2013)
10. Naor, M.: Verification of a human in the loop or identification via the turing test (1996)
11. Nielsen, F.Å.: A new anew: Evaluation of a word list for sentiment analysis in microblogs. CoRR, abs/1103.2903 (2011)
12. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
13. Vikram, S., Fan, Y., Gu, G.: Semage: A new image-based two-factor captcha. In: Proceedings of the 27th Annual Computer Security Applications Conference, ACSAC 2011, pp. 237–246. ACM, New York (2011)
14. Warner, O.: Kittenauth (2009), `http://www.thepcspy.com/kittenauth`
15. Yamamoto, T., Suzuki, T., Nishigaki, M.: A proposal of four-panel cartoon captcha. In: 2011 IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 159–166 (March 2011)
16. Zhu, B.B., Yan, J., Li, Q., Yang, C., Liu, J., Xu, N., Yi, M., Cai, K.: Attacks and design of image recognition captchas. In: Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, pp. 187–200. ACM, New York (2010)

Regression from Distributed Data Sources Using Discrete Neighborhood Representations and Modified Stalked Generalization Models

Héctor Allende-Cid, Claudio Moraga, Héctor Allende, and Raúl Monge

Abstract. In this work we present a Distributed Regression approach, which works in problems where distributed data sources may have different contexts. Different context is defined as the change of the underlying law of probability in the distributed sources. We present an approach which uses a discrete representation of the probability density functions (pdfs). We create neighborhoods of similar datasets, comparing their pdfs, and use this information to build an ensemble-based approach and to improve a second level model used in this proposal, that is based in stalked generalization. We compare the proposal with other state of the art models with 5 real data sets and obtain favorable results in the majority of the datasets.

Keywords: Distributed Machine Learning, Context-aware Regression, Similarity representation.

1 Introduction

Automatic learning has become increasingly important over the years because of the rapid growth of the amount of data that is being stored. In the last years, the total amount of information available on the Internet has had a exponential growth. By 2005, the total size was around 600 terabytes [9]. Nowadays, the total amount of data is almost incalculable. This rapid growth of the data available, presents new

Héctor Allende-Cid · Héctor Allende · Raúl Monge
Departamento de Informática, Universidad Técnica Federico Santa María
Avenida España 1680, Valparaíso, Chile
e-mail: vector@inf.utfsm.cl

Claudio Moraga
European Centre for Soft Computing, 33600, Mieres, Asturias, Spain

Claudio Moraga
TU Dortmund University, 44220 Dortmund, Germany

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_27

oportunities for applications of Machine Learning and Automatic Data Analysis. Since human reasoning is not able to handle large data sets, the need of automatic data analyzers is a necessity. Due to this reason, challenges related with the scalability and efficiency of the learning algorithms have become of central importance. Classic machine learning algorithms, usually work with monolithic data sets, this means that the entire data set is loaded into main memory. When the amount of data is very large, this is unfeasible, because the algorithm will not be able to train on the whole data set, or it will be impractical due to computational or memory restrictions. To handle this type of problems, Parallel and Distributed approaches are having a lot of attention from the Machine Learning community. Distributed Machine Learning (DML) is often mentioned with Parallel Machine Learning (PML) in literature. While both attempt to improve the performance of traditional Data Mining systems, they assume different system architectures and take different approaches. In DML, computers and data are distributed and communicate through message passing. In PML, a parallel system is assumed with processors sharing memory and/or storage. Computers in a DML system may be viewed as processors sharing nothing. This difference in architecture greatly influences algorithm design, cost model, and performance measure in distributed and parallel Machine Learning.

The field of Distributed Machine Learning (DML) has been very active and is enjoying a growing amount of attention since it was first proposed. There are many real world applications where the data is distributed naturally. In most cases, the amount of data distributed is so large, that is unfeasible to trasmit it to a centralized node, so there is no alternative other than to treat the problem with a Distributed Learning approach.

Most of the current DML techniques treat the distributed data sets as a single virtual table and assume that there is a global model which could be generated if the data were combined or centralized, completly neglecting the different semantic contexts that this distributed data sets may have [26]. If we see this as a statistical learning problem, we deal with samples of data that follow different underlying laws of probability. Loosely speaking Machine Learning models try to find a function which relates a given output $\underline{y}$ with an input vector $\underline{x}$. The classic Machine Learning approach is related with the estimation of the joint probability distribution $H(\underline{X}, \underline{Y})$. The joint probability distribution can be decomposed in the conditional probability distribution and the marginal one ($H(\underline{X}, \underline{Y}) = F(\underline{Y}/\underline{X})G(\underline{X})$). In this paper, context is defined as the joint probability distribution that governs each data source.

The next section will present a brief view of the state of the art related to this work. In section 3 we present the proposed algorithm which will be able to address the problem presented above. In section 4 we show some experimental results with real datasets. The last section is devoted to discuss the results and to make some conclusions.

2 State of the Art

There is a large number of works proposed in the last decade in the field of Distributed Machine Learning. A large fraction of DML algorithms focuses on combining predictive models. This approach has emerged from empirical experimentation due to a requirement for higher prediction accuracy. Recently, several researchers treat distributed learning systems as a centralized ensemble-based method [8, 14, 15, 24]. Several learning algorithms are applied at each local site, using separate training data to mine local knowledge. A new data point is then classified/predicted from the predictions of all local sites using ensemble methods such as stacking, boosting, majority voting, simple average, or winner-takes-all methods. In general, DML approaches apply ensemble methods to minimize the communication costs and to increase the performance of the system predictions.

There are well known approaches for distributed classification problems (see e.g. [6, 18, 20, 22, 24] . It is not straightforward to adapt this approaches to the regression task. Both problems are related, but the strategies to solve them are different. Another drawback of these methods, is that they assume that the underlying laws of probability of the distributed sources are the same. This is an assumption that is inherited from the ensemble-based approaches in classic machine learning. When the distributed is resampled, it is assumed that the resampled data follows the same underlying law of probability as the original set. In real-world distributed problems, it is impossible to assure that, because the real underlying law of probability is unknown.

Another challeging task is the one of Distributed Data Clustering. In [4] the authors present a distributed k-means and k-median approach in topologies of general communication. In [10] the authors present a generic distributed data clustering algorithm applied to sensor networks. Other distributed clustering works that can be found recently in the literature address the problem of larga-scale data sets and evolving data streams [11–13]. There are also distributed approaches to other tasks or problems, e.g. Genetic Algorithms [16].

The task that will be addressed in this work is the regression task. Loosely speaking the task of regression can be presented as the task of finding the relationship between input and output variables, where the outputs are real-valued. This can be used for predicting an expected value by presenting to the model an unknown input. Works related with the regression task in Distributed Machine Learning are scarce. The majority of algorithms deal with the classification problem. In this work the task of regression will be addressed.

Yan Xing et al. [28, 29] have proposed a series of algorithms based on, what the authors call, a meta-learning approach to deal with the regression problem addressing the context heterogeneity case. The authors propose a meta-learning-based hierarchical model that is able to be successfully used in distributed scenarios with context heterogeneity. The definition of context in this work is the variance that the distributed sources have in their outputs, thus neglecting the context change in the input space. The authors claim that this change of context between distributed sites is random.

In [1], an ensemble approach based on building neighborhoods of similar datasets is presented. To build the neighborhoods, it is assumed that the datasets follow a known underlying law of probability, and using the Hypothesis Test based on divergence measures, they form the corresponding neighborhoods. In this work it is necessary to assume a known underlying law of probability, i.e. multivariate normal distribution, in order to perform the Hypothesis tests [19, 23]. The algorithm can be summarized as follows: At first, local models are trained with the available distributed data sets. In each distributed node, a local algorithm is trained with its corresponding data. After that, assuming that the underlying laws of probability of each distributed data sets are known (multivariate gaussian distributions), the mean and variance-covariance matrices (parameters of the underlying law of probability) are shared across all distributed nodes. With this information, in each distributed node i, an Hypothesis test is performed with the parameters of node i and the parameters all other nodes $j = 1, \ldots, k$, where $j \neq i$ and k is the total number of distributed data sets. After performing all Hypothesis tests, the neighborhood for node i is built if there was no evidende to reject H_0 (that the parameters of both underlying laws of probability were the same). Also all the local models are shared across the sites. Then, a second stage learner is trained, where the inputs of this learner are the outputs of *all* local models. The final output of the model is the ensemble of all second stage models, that belong to the same neighborhood, where the new data inputs are registered.

For a more complete review on the state of the art of Distributed Machine Learning Algorithms, please refer to [21].

3 Proposal

To avoid working under the assumption of a known underlying law of probability, we can use a discrete representation of the probability density functions (pdfs). For this we can use n-dimensional histograms. A histogram $H(x)$ of a set $[\underline{x}_1, \underline{x}_2]$ represents the frequency of each value. If the dataset is one-dimensional, the histogram is represented by a vector. The length of the vector depends in the number of bins used to represent the histogram. If the dataset is 2-dimensional, the histogram is represented by a matrix. For n-dimensional cases the data structure used to represent is a n-rank tensor.

Before building histogram representations for all distributed datasets, the minimum/maximum per input dimension must be shared across all data sets. This is necessary, because we need to build histograms that are comparable with each other. To make them comparable we need bins with the same limits. This information shared across the system, does not contradict the restriction of sharing raw data, because it only shares the min and max global values of the examples of each distributed source. With this data, we obtain the global min and max values of all distributed sources, using this information to build histograms for all distributed sources, with the same bin limits. The idea is to use a histogram representation to build a vector of size r,

that represents the dissimilarity between 2 datasets, using r distance measures. We define a number of different distance/similarity measures, using preferable distances from different families [7]. If 2 data sets are similar the distance vector will be near (0,0,..., 0). If there are k distributed data sources, distances vectors $\{d_{i1}, \ldots, d_{ik}\}$ will be generated for node i. Applying a clustering algorithm to all distance vectors, we can define a neighborhood for node i, by taking into account all distance vectors that belong to the same cluster of d_{ii}. E.g. if the distance vector d_{i1} and d_{i3} of a total of $k = 5$ distributed nodes, then the nodes 1 and 3 belong to the neighborhood of i. Preliminary results regarding histogram representation were presented in [3]. The main difference between this approach and the one presented in [3] is that the second level models, in this proposal are trained only with the outputs of the local models, that belong to the same neighborhood. In the previous work, the second level models were trained with all the outputs of the local models, wether they belonged to the neighborhood or not. In this work the clustering algorithm used is the Hierarchical Clustering algorithm.

In the next subsection we present the proposed algorithm:

3.1 Learning

1. *Phase 1 - Local Learning*. Suppose that there are k distributed data sets. At each node N_i, where $i = 1, \ldots k$, we use an available learning algorithm to train a local predictive model L_i from data source D_i of that node. The choice of the learning algorithm is not restricted to any particular kind. The local training data consists in an n dimensional input vector $((x_{i1}, x_{i2}, \ldots, x_{in}))$ and a response variable y_i.
2. *Phase 2 - Model and information transmission*. Each node N_i, where $i = 1, \ldots, k$ receives the minimum and maximum of each attribute of the other nodes. With this information each distributed node N_i creates a n-dimensional histogram. The histograms are then shared across the distributed nodes. Then in each distributed node, the distance vector is calculated, with respect to all other nodes. Also the parameters of all the local models are shared across the distributed nodes, so each node has a copy of the rest of the local models. With a clustering algorithm, we generate the neighborhoods of distributed nodes. The result is a binary vector (h_i with k binary variables, which indicates the nodes which follow the same underlying law of probability of D_i. In other terms we will refer to this vector as the Neighborhood vector. E.g. If there are 5 distributed nodes D_1,D_2,D_3, D_4 and D_5, and we are checking if node D_1 has the same distribution as the rest, and only the variables $h_1(1)$, $h_1(2)$ and $h_1(4)$ are equal to 1, meaning that the data contained in the nodes N_2 and N_4 follow the same distribution as in node N_1.
3. *Phase 3 - Generation of second level learners*. Since every node N_i has a copy of the other local models, each local model L_l, $l = 1, \ldots, k$, contained in node N_i is trained with the local data from that node (D_l). Each of the local models that belong to the neighborhood of node N_i outputs a response variable $\hat{y}_{ij}$ with the local data D_i, where $j = 1, \ldots, k$. Each local node applies then a stacked

learning algorithm G_i (second level model) which is trained with the outputs of local models $(\hat{y}_{ij})$, where j indicates the nodes that belong to the neighborhood of N_i and the real response variable y_i, obtained from the training data of the node D_i. This is inspired in the stacking model of Wolpert [27].

3.2 Predicting

1. *Final output of the proposed model*. The output of our model is the following: Whenever a new example arrives at a node N_i, we compute all the outputs of the local models that are stored in this node. We have an apriori information of which of the other nodes have data following a close underlying law of probability of the current node, which is reflected in the neighborhood vector mentioned above (h_j, where $j = 1, \ldots, k$). Then the output of the local models in this node are transmitted to only the other nodes which have a non-zero label in this vector. The final output of the model is the weighted sum of all the G_i model outputs that received the output of the local models of their corresponding neighborhoods h_i. The weights are calculated with the following equation: $w'_j = \frac{w_j}{\sum w_j}$, where $w_j = 1 - |d_{ij}|^2$. E.g. in the example presented in Fig. 1 the final output of the model is the weighted average of models G_1, G_3 and G_k, because only the variables $h_2(1)$, $h_3(3)$ and $h_3(k)$ are distinct from zero.

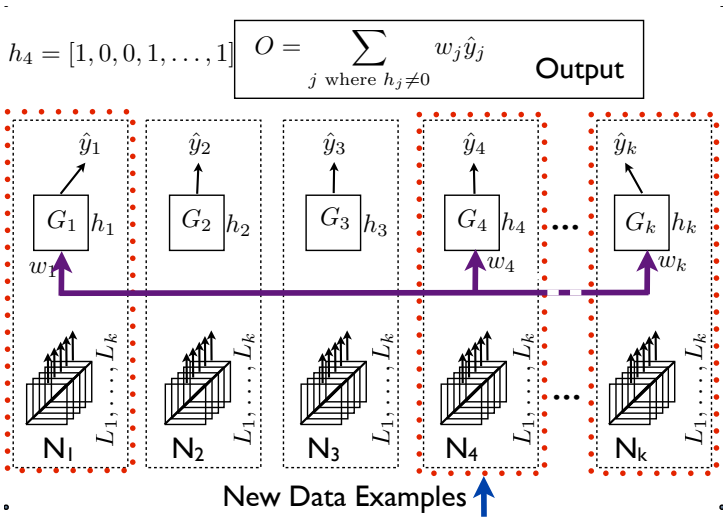


Fig. 1 Architecture of the proposed system

4 Experimentation

In this section we present the results obtained by our proposal with 5 real world data sets in terms of Mean-squared error (MSE). The Communities and Crime, Parkinson and Bike Sharing and Wine Quality data sets can be obtained from [18]. The Communities and Crime dataset consists in 1994 examples and 128 attributes. The data was split in 32 datasets. The Parkinson dataset consists in 5875 examples and 26 attributes. It was split in 42 datasets, according to the patients ID. The Bike Sharing dataset consiste in 17389 examples and has 16 attributes. It was split in 12 datasets, according to the month of the year. The Wine dataset consists in 1599 examples of red wine and 4898 examples of white wine. The number of attributes is 11. The latter one was split in 10, 20 and 30 datasets, where half of the datasets are generated from the red wine examples and the other half from the white wine examples. The Wind dataset consists in 29050 examples and was distributed in 54 sources. For further details of the Wind dataset, please refer to [2].

To compare our proposal we tested 4 different algorithms:

- Reference model that has access to all data. (*Global*)
- Model presented in [29]
- Model presented in [1]
- Proposal.

The performance measure is the mean value of the MSE obtained in each distributed source. This means that in each distributed source 80% of the data was set for training purposes and the other 20% for testing purposes. In each node the MSE was obtained with the testing data. The final performance measure is the mean of the MSE results obtained in each node. The local and the second level models were feed-forward artificial neural networks. The optimum number of neurons of the hidden layer was searched from 2 to 10 neurons. The clustering algorithm used in this experimentation to create the neighborhoods was the Hierarchical Clustering. The distance metrics used were the Euclidean, Sorensen, Intersection and Kullback-Leibler distances.

The results of our proposal are presented in Table 1. The Mean and standard deviation of 20 experimental runs are presented. The performance measure is the

Table 1 Mean value and standard deviation of 20 experimental runs

Dataset	# Sources	Global	[29]	[1]	Proposal
Parkinson	48	0.0022 ± 0.0004	**0.0025 ± 0.0002**	0.0034 ± 0.0006	0.0033 ± 0.0006
Crime	32	0.0080 ± 0.0031	**0.0335** ± 0.0057	0.0380 ± 0.0063	0.0363 ± **0.0056**
Bike	12	1.3759 ± 2.7726	13.660 ± 16.098	59.033 ± 15.812	**1.5803 ± 0.592**
Wind	54	30.661 ± 5.6572	42.936 ± 2.282	65.675 ± 9.318	**35.439 ± 2.574**
Wine	10	0.6014 ± 0.0253	0.8278 ± 0.1707	0.6268 ± 0.0254	**0.5406 ± 0.0131**
Wine	20	0.6025 ± 0.0238	0.8378 ± 0.1223	0.9258 ± 0.0617	**0.5293 ± 0.0165**
Wine	30	0.5998 ± 0.0132	0.8375 ± 0.077	1.1530 ± 0.0751	**0.5244 ± 0.0132**

Mean-squared error. We compare the results with the models presented in [29] and [1]. The results were favorable for our proposal in the majority of the presented datasets. There were only 2 datasets (Parkinson and Crime), where the model proposed in [29] outperformed our model. Despite that, the difference between our proposal and [29] are not that large. In the rest of the presented datasets, the proposal outperformed the other approaches. As can be seen in the table, the difference was in some cases considerable. The Parkinson dataset, has been always been treated as a monolithic dataset in many works, so it is understandable that there is no considerable difference between the examples, thus affecting the proposal. It also should be pointed out, that the proposal outperforms the previous approach presented in [1] in every dataset. The results of the Global model, a model that uses all the data centralized, are only reported for comparison purposes.

5 Conclusions

In this proposal we present a distributed regression approach that is able to detect different contexts in the input space, thus improving the performance of local models in the task of regression from distributed sources. As the results show, it is very important, not to neglect the different contexts that are present in the distributed sources. Also it is sometimes impractical to assume that the underlying law of probability is known, so a discrete way of representing it is necessary. Also it is important to focus on the second level models, that are based on stalked generalization, in order to filter the inputs they receive (receive inputs only when they are part of the neighborhood). The clustering algorithm is also crucial, so in future studies, different algorithms like the found in [5] or [17] could improve the results. Further studies are necessary in order to establish error bounds and to formally prove the convergence of the algorithm.

Acknowledgements. This work was supported by the following research grants: Fondecyt 1110854 and DGIP-UTFSM. The work of C. Moraga was partially supported by the Foundation for the Advancement of Soft Computing, Mieres, Spain and by the CICYT Spain, under project TIN 2011-29827-C02-01.

References

1. Allende-Cid, H., Moraga, C., Allende, H., Monge, R.: Context-Aware Regression from Distributed Sources. In: IDC 2013, Prague, Czech Republic, pp. 17–22 (2013)
2. Allende-Cid, H., Moraga, C., Allende, H., Monge, R.: Wind Speed Forecast under a Distributed Learning Approach. In: V Chilean Workshop of Pattern Recognition, Temuco, Chile (2013)
3. Allende-Cid, H., Allende, H., Monge, R.: Soft Computing applied to Distributed Regression with Context-Heterogeneity. Submitted to the Journal of Multivalued Logic and Soft Computing (January 2014)

4. Balcan, M.-F., Ehrlich, S., Liang, Y.: Distributed k-means and k-median clustering on general communication topologies. Paper presented at the meeting of the NIPS (2013)
5. Bello-Orgaz, G., Menéndez, H., Camacho, D.: Adaptive K-Means Algorithm for overlapped graph clustering. International Journal of Neural Systems 22(5), 1–19 (2012)
6. Caragea, D., Silvescu, A., Honavar, V.: Analysis and synthesis of agents that learn from distributed dynamic data sources. In: Wermter, S., Austin, J., Willshaw, D.J. (eds.) Emergent Neural Computational Architectures Based on Neuroscience, pp. 547–559 (2001)
7. Cha, S.-H.: Comprehensive survey on distance/similarity measures between probability density functions. International Journal of Mathematical Models and Methods in Applied Sciences 1(4), 300–307 (2007)
8. Chawla, N.V., Lawrence Hall, O., Kevin Bowyer, W., Phillip Kegelmeyer, W.: Learning ensembles from bites: A scalable and accurate approach. Journal Machine Learning Res. 5, 421–445 (2004)
9. D-Lib Magazine. A research library based on historical collections of the Internet Archive (2000), http://www.dlib.org/dlib/february06/arms/02arms.html (accesed February 26, 2014)
10. Eyal, I., Keidar, I., Rom, R.: Distributed data clustering in sensor networks. Distributed Computing 24(5), 207–222 (2011)
11. Forman, G., Zhang, B.: Distributed data clustering can be efficient and exact. SIGKDD Explor. Newsl. 2(2), 34–38 (2000)
12. Hefeeda, M., Gao, F., Abd-Almageed, W.: Distributed approximate spectral clustering for large-scale datasets. In: Proceedings of the 21st International Symposium on High-Performance Parallel and Distributed Computing, HPDC 2012 (2012)
13. Ienco, D., Bifet, A., Zliobaite, I., Pfahringer, B.: Clustering Based Active Learning for Evolving Data Streams. Discovery Science, 79–93 (2013)
14. Lattner, A., Grimme, A., Timm, I.: An evaluation of Meta Learning and Distributed Strategies in Distributed Machine Learning. In: European Conference on Data Mining 2010, pp. 67–74 (2010)
15. Lazarevic, A., Obradovic, Z.: The Distributed Boosting Algorithm. In: Knowledge Discovery and Data Mining, pp. 311–316 (2001)
16. López, L.I., Bardallo, J.M., De Vega, M.A., Peregrin, A.: Regaltc: A distributed genetic algorithm for concept learning based on regal and the treatment of counter examples. Soft Comput. 15(7), 1389–1403 (2011)
17. Menéndez, H., Barrero, D., Camacho, D.: A Genetic Graph-based approach for Partitional Clustering. International Journal of Neural Systems 24(1430008), 1–19 (2014)
18. Moretti, C., Steinhaeuser, K., Thain, D., Chawla, N.V.: Scaling up classifiers to cloud computers. In: Proceedings of the 8th IEEE International Conference on Data Mining (ICDM), pp. 472–481 (2008)
19. Pardo, L.: Statistical Inference Based on Divergence Measures. Ed. Chapman and Hall (2005)
20. Park, B., Kargupta, H.: Distributed Data Mining: Algorithms, Systems, and Applications. In: Data Mining Handbook (2002)
21. Peteiro-Barral, D., Guijarro-Berdinas, B.: A survey of methods for distributed machine learning. Journal of Progress in Artificial Intelligence 2, 1–11 (2013)
22. Rodríguez, M., Escalante, D.M., Peregrín, A.: Efficient distributed genetic algorithm for rule extraction. Appl. Soft Comput. 11(1), 733–743 (2011)
23. Salicrú, M., Morales, D., Menéndez, M.L., Pardo, L.: On the applications of divergence type measures in testing statistical hypotheses. J. Multivar. Anal. 51(2), 372–391 (1994)
24. Tsoumakas, G., Vlahavas, I.P.: Effective Stacking of Distributed Classifiers. In: ECAI 2002, pp. 340–344 (2002)
25. Bache, K., Lichman, M.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine (2013), http://archive.ics.uci.edu/ml
26. Wirth, R., Borth, M., Hipp, J.: When distribution is part of the semantics: A new problem class for distributed knowledge discovery. In: ECML 2001, pp. 3–7 (2001)

27. Wolpert, D.: Stacked Generalization. Neural Networks 5(2), 241–259 (1992)
28. Xing, Y., Madden, M., Duggan, J., Lyons, G.: Context-based Distributed Regression in Virtual Organizations. In: Parallel and Distributed Computing for Machine Learning. 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2003), Cavtat-Dubrovnik, Croatia (2003)
29. Xing, Y., Madden, M.G., Duggan, J., Lyons, G.J.: Context-Sensitive Regression Analysis for Distributed Data. In: Li, X., Wang, S., Dong, Z.Y. (eds.) ADMA 2005. LNCS (LNAI), vol. 3584, pp. 292–299. Springer, Heidelberg (2005)

On a Machine Learning Approach for the Detection of Impersonation Attacks in Social Networks

Esther Villar-Rodriguez, Javier Del Ser, and Sancho Salcedo-Sanz

Abstract. Lately the proliferation of social networks has given rise to a myriad of fraudulent strategies aimed at getting some sort of benefit from the attacked individual. Despite most of them being exclusively driven by economic interests, the so-called impersonation, masquerading attack or identity fraud hinges on stealing the credentials of the victim and assuming his/her identity to get access to resources (e.g. relationships or confidential information), credit and other benefits in that person's name. While this problem is getting particularly frequent within the teenage community, the reality is that very scarce technological approaches have been proposed in the literature to address this issue which, if not detected in time, may catastrophically unchain other fatal consequences to the impersonated person such as bullying and intimidation. In this context, this paper delves into a machine learning approach that permits to efficiently detect this kind of attacks by solely relying on connection time information of the potential victim. The manuscript will demonstrate how these learning algorithms – in particular, support vector classifiers – can be of great help to understand and detect impersonation attacks without compromising the user privacy of social networks.

Keywords: Impersonation, Social Networks, Support Vector Machines.

1 Introduction

Since the late 90's when the first social networking sites were founded and started (e.g. Friendster, Friends Reunited or Six Degrees), these networks have become an essential tool for people to express their feelings, strike up new relationships or keep

Esther Villar-Rodriguez · Javier Del Ser
TECNALIA. OPTIMA Unit, E-48160 Derio, Spain
e-mail: {esther.villar,javier.delser}@tecnalia.com

Sancho Salcedo-Sanz
Universidad de Alcalá, 28871 Alcalá de Henares, Madrid, Spain
e-mail: sancho.salcedo@uah.es

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_28

in contact with close yet physically distant friends. According to recent statistics published by Pew Internet in [1], 73% of on-line adults between 30 and 49 years old use social networking sites, rising up to a meaningful 90% if the focus is placed on the 18-29 age range. These factual indicators elucidate the relevance and impact of social networking in regards not only to current methods and procedures for socialization, but also to criteria for decision making in fields such as Telecommunications [2], Health [3] and Education [4], among others.

However, these networks are also regarded as an appealing and profitable substrate for committing illegal acts or cybercrimes. Reasons for this abound, and span far beyond the popularity of social media and the addiction and lure it causes over its users: social media has become an ideal place for crime also due to the noted transition from emails to social media, their growing usage in smart-phones and the veracity and sensitivity of the information exchanged through this communication channel. This rationale has certainly ignited the number of attacks to social platforms, as evinced by the unauthorized access to the details of approximately 250.000 Twitter users in early 2013 [5].

This paper is focused on one of the most frequent attacks in social media platforms, in which someone gains unauthorized access to another's account by stealing his/her credentials or by creating a fake profile, which results in the victim being impersonated. The purpose of this attack can be discriminated depending on the interactivity required from the attacker for its achievement. On the one hand interactive impersonation refers to those attacks in which the victim's account is exploited to send – or post – bulk messages with an intentionally malicious content. Most of the literature related to this first category gravitates on detecting phishing, masquerading or other attacks by creating fully dedicated systems to analyze specific characteristics of the bulk messages sent/left by the impostor [6,7]. Another less-intrusive detection approach to deal with this family of attacks hinges on inferring behavioral profiles of the user interaction with the platform and subsequently discriminating non-typicalities that may correspond to an unauthorized usage of his/her account. The state of the art regarding behavioral profiling in social networks is still in its early stages, being the recent work by Egele et al in [8] the first attempt to address compromised accounts via this method. In this work the detection of this attack is analyzed from a general, theoretical standpoint by explicitly mentioning the concept of behavioral patterns to distinguish illegitimate uses. However, the contribution of the manuscript finally concentrates on the messages as the main resource to be analyzed for detecting the attack.

The other category in which impersonation over social networks can be classified does not involve any interactivity of the attacker with the platform. It is often the case – specially among teenagers – that the attack is stealthy since its purpose is merely to capture personal or sensitive information about the user. In fact, a very high percentage of the actual use of social networks by young users is gossiping, i.e. multitude of connections to the platform in which no interaction is performed, but which are representative of the degree of *dependency* of the user towards the application. The detection of this attack results to be extremely complex under these non-interactivity constraints, and surprisingly has not been addressed in the litera-

ture even though the upsurge of mobile applications, tablets and other lightweight devices have sparked much more frequent connections. This connectivity statistics (duration, frequency, periodicity) can be of great help to non-intrusively characterizing the usage of the platform of a given user, based on which the manager of the social network can preemptively flag possible impersonation attacks with more or less meaningfulness depending on the connection erraticism of the user under analysis. Scarce contributions can be found in the literature dealing with this subtle attack [8, 9], which exclusively resort to the regularity of the social graph of the user without any consideration to the timing of his/her connections to the platform.

In this context, this manuscript presents an impersonation detection system that builds upon three different processing stages, each leveraging information from the user with different privacy implications. In this preliminary work the first processing step of such a system will be described in detail, which operates exclusively on connection time information. The goal of this first stage is not to firmly claim that an impersonation attack has been held, but to trigger subsequently, more sophisticated albeit privacy-intrusive phases processing the social graph of the user (via e.g. dynamic link grouping or centrality metrics) or the content itself (by resorting to e.g. semantic and natural language processing). To this end, this early-warning stage 1) transforms the connection time information to a feature space yielding more condensed multidimensional profiles for the user under consideration; 2) trains a binary support vector machine (SVM [10]) classifier with the available feature history and synthetically yet realistic connection traces of potential attackers; and 3) estimates the false alarm and detection probabilities that reflect its performance.

2 Problem Formulation

The detection of impersonation attacks in the first stage presented in this paper can be mathematically modeled as follows: let the user under analysis be labeled as A, with connection times compiled in the time-variant vector $\mathbf{w}_t^A \triangleq \{w_{t,1}^A, \ldots, w_{t,N}^A\} = \{w_{t,n}^A\}_{n=1}^N$. Index variables t and n depend on the granularity for which such connection time statistics are captured by the detection tool: for instance, if connection duration per hour is captured every day, t would represent the day index, whereas n would indicate the hour index with $N = 24$ if the recording is made all day and night long. As in any other binary detection process, the problem of spotting impersonation attacks reduces to the testing of two mutually exclusive hypotheses, namely

$$\mathcal{H}_0\text{: user } A \text{ has NOT undergone any impersonation attack,} \tag{1}$$

$$\mathcal{H}_1\text{: user } A \text{ has undergone an impersonation attack,} \tag{2}$$

which, as aforementioned in the introduction, have to be tested by solely utilizing the connection time information stored in $\mathbf{w}_t^A$. This information can be collected during a certain period of length T^* which, for the sake of notational ease, will be denoted as the matrix $\mathbf{W}_{T^*}^A \triangleq \{\mathbf{w}_t^A\}_{t=1}^{T^*}$. It is intuitive to note that the above two

hypotheses must also inherit the time dependency of the information used for their testing to yield

$$\mathcal{H}_0^{T^*}\text{: user } A \text{ has NOT undergone any impersonation attack at time } T^*, \quad (3)$$

$$\mathcal{H}_1^{T^*}\text{: user } A \text{ has undergone an impersonation attack at time } T^*, \quad (4)$$

from which the false alarm and detection probabilities can be expressed as $P_{fa}^{T^*} \triangleq Pr\{\mathcal{H}_1^{T^*}|\mathcal{H}_0^{T^*}\}$ and $P_d^{T^*} \triangleq Pr\{\mathcal{H}_1^{T^*}|\mathcal{H}_1^{T^*}\}$, respectively.

From an algorithmic point of view this hypothesis test can be understood as a binary classification problem whose goal is to learn to classify correctly two types of classes: 0 (`no attack`) and 1 (`attack`). This can be accomplished by the mapping $f : \mathbf{w} \mapsto y$, where $\mathbf{w}$ stands for the variable corresponding to the connection times and y a binary variable representing the class (`no attack` or `attack`) to which vector $\mathbf{w}$ is mapped. This mapping can be learned from the record of past connection times $\mathbf{W}_{T^*}^A$ under the assumption that no attack has occurred within the T^* time frame. If no counter-examples for possible impostors are available for the mapping learning process, two different strategies can be followed:

A. To utilize only $\mathbf{W}_{T^*}^A$ for the training process, which eventually leads to a so-called one-class classifier.
B. To generate synthetic connectivity traces for possible impostors based on expert knowledge on the casuistry of this kind of attacks, which ultimately produces a binary classifier.

In this paper strategy B will be adopted, the reason being that in absence of training samples for class 1 (`attack`), one-class classifiers may underfit the $\mathbf{w}$-space corresponding to the user A under analysis. This ultimately leads to a decreased probability of detection due to impersonation being incorrectly labeled as 0 (`no attack`). The manuscript will later delve into how this synthetic attack modeling has been tackled.

Once a model for $f(\mathbf{w})$ has been constructed, the performance of the resulting classifier can be quantified in terms of the error function $E^{T^*}(\mathbf{w}, y)$, which relates to the aforementioned detection indicators as

$$E^{T^*}(\mathbf{w}, y) = \begin{cases} 1 \text{ if } \begin{cases} \mathbf{w} \text{ belongs to } A \text{ and } f(\mathbf{w}) = 1 \text{ (attack)} \bigstar \\ \mathbf{w} \text{ does not belong to } A \text{ and } f(\mathbf{w}) = 0 \text{ (no attack)} \end{cases} \\ 0 \text{ if } \begin{cases} \mathbf{w} \text{ belongs to } A \text{ and } f(\mathbf{w}) = 0 \text{ (no attack)} \\ \mathbf{w} \text{ does not belong to } A \text{ and } f(\mathbf{w}) = 1 \text{ (attack)} \clubsuit \end{cases} \end{cases}$$

from which $P_{fa}^{T^*}$ and $P_d^{T^*}$ can be computed by accounting for the classification events marked with $\bigstar$ and $\clubsuit$ in the above expression, respectively. The problem is then to find a classifier that simultaneously maximizes $P_d^{T^*}$ and minimizes $P_{fa}^{T^*}$ $\forall T^*$, which can be instead reformulated as the maximization of the area under the ROC (Receiver Operating Characteristic) curve depicting the $(P_d^{T^*}, P_{fa}^{T^*})$ pairs of the detector for different values of T^*.

3 Proposed User Profiling Approach

Connection time information reflect daily patterns in users' habits and may serve as early indicators of a behavioral change, from e.g. users who never connect within their working hours to those who generate short connections scattered throughout the day (except sleep hours). Fortunately, inferring the time connectivity habits of potential victims results to be extremely difficult for impostors, thus they decide to attack users with very sparse connections to avoid being intercepted.

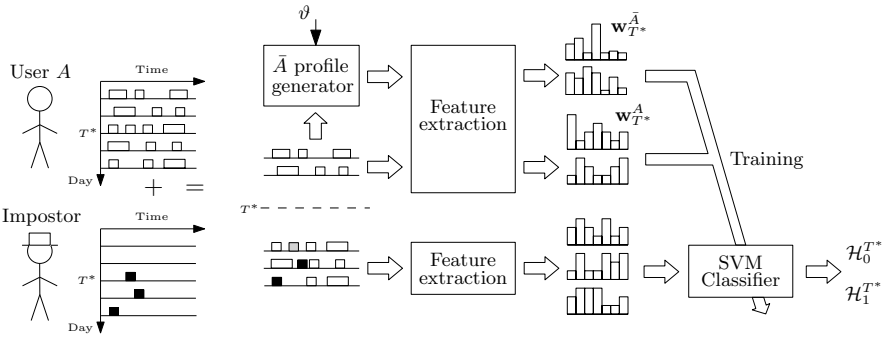


Fig. 1 Proposed connection time based detector of impersonation attacks

The proposed approach leverages this assumption in order to construct the connection-based system for impersonation detection depicted in Figure 1. Connection time information for user A is captured at a certain sampling granularity. In this context, such connection time information can be quantified as the number of daily connections, hours of the day at which such connections have taken place, their duration, etc. In order to determine if a behavior regarding connections is unknown according to the learned patterns, enough sampling granularity is necessary to feed the system with meaningful data to properly characterize the connection behavior of the user, but it has to be also permissive enough to accommodate eventual behavioral changes of the user that should not be attributed to a potential impersonation attack.

Bearing this trade-off in mind the proposed scheme opts for hourly aggregated connection information: a coarser granularity such as aggregated data per day could hinder the detection of an unauthorized access if the user connects in a daily basis. By contrast, a finer sampling rate for this information might entail posterior over-fitting issues in the classifier and consequently, less flexibility in the system to properly classify minor behavioral perturbation of the user. In conclusion, the relevant time information for characterizing the behavior of a social network user lies in the frequency of connections and their shape: how many times a user connects per hour, how much each connection lasts, how many minutes long the longest session is, etc.

However, the consideration of an hour as the granular unit for the features to be input to the classifier could imply that an one-hour deviation in the connection habits of the user would produce a false alarm. Intuitively it cannot be assumed that a user

will be regular enough to maintain a strict daily connection schedule with less-than-one-hour inter-day deviations. Based on this rationale, the connection time information captured from user A is transformed to an alternative feature space with a two-fold aim: 1) to reduce the computational complexity of the subsequent classification stage when processing the resulting high-dimensional feature space; and 2) to prevent the system from necessitating very large datasets due to the so-called course of dimensionality and the need for a properly balanced under- (low probability of detection) and over-fitted (high probability of false alarm) classifier.

Therefore hourly captured connection time information is mapped to a smaller feature space representing the most relevant information within a 24 hours time frame to characterize the behavior of the user. This new feature space aggregates the captured time statistics of the user into periods corresponding to morning, noon, evening and night, which are intuitively of meaningfulness for the behavioral characterization purposes of the platform, i.e. a user is assumed to follow a certain regularity within such periods. Given the notation in Section 2 the new feature space is defined by $\mathbf{w}_t^A \triangleq \{w_{t,n}^A\}_{n=1}^N$ with $N = 10$ and the following entries:

- Overall duration of connections in the morning (07:01-13:00).
- Overall duration of connections during lunchtime (13:01-17:00).
- Overall duration of connections in the evening (17:01-0:00).
- Overall duration of connections at night (0:01-07:00).
- Number of hours with at least one connection in the morning.
- Number of hours with at least one connection during lunchtime.
- Number of hours with at least one connection in the evening.
- Number of hours with at least one connection at the night.
- Mean duration averaged over the longest eight daily connections.
- Median of the duration of all connections within the day.

Another objective of using this new feature space is to potentially model the behavior of any user of the social network, but respecting the peculiarities which could distinguish him/her from an eventual impostor. In other words, the goal is to infer a model capable of representing different patterns in a precise and consistent fashion that discriminates efficiently sibylline identity thefts. This is accomplished by generating a multi-dimensional classification model with compact and cohesive clusters with empty space in between where a moderate impersonation could be hidden. Once this feature extraction has been performed, it is expected that the produced set of features discriminates better false alarms and detects true attacks in a more reliably manner than by using the hourly statistics mentioned before.

3.1 Selection of the Classifier

To the knowledge of the authors this manuscript embodies the first practical approach to the detection of impersonation attacks in social networks. Consequently, a brief survey on similar earlier work done for other kind of attacks has been done. Due to

their interactive nature, in general phising attacks are detected by resorting to textual features, subsequently fed to machine learning techniques or statistical methods [11, 12]. Among them, Support Vector Machines (SVM) have been empirically shown to be one of the most effective methods certified by their outperforming properties with respect to other classifiers, although Artificial Neural Networks (ANN), Self Organizing Maps (SOMs) and other algorithms have been also applied to this paradigm with satisfactory results [13]. Bearing this in mind, we decided to adopt the SVM technique with a radial basis function (RGB) kernel which also avoids assuming other statistical distributions applied for example in k-means or this kind of clustering algorithm. SVM classifier will permit adjust the model to the patterns creating empty spaces amongst the positive examples where a subtle attack could be taken place (trying to enhance the detection capability of the algorithm).

In regards to its training process the SVM classifier has been fed with the transformed feature set $\mathbf{w}_t^A$ corresponding to the connection time information of the user under consideration. Furthermore, as shown in Figure 1 and argued in Section 2, the training procedure uses a second set of synthetically generated connection traces created independently based on – and balanced in number with – the former one. This second set models the complementary space corresponding to potential impersonation attackers, and is furnished by scaling the connection time traces of user A considering the mean and standard deviation of the set of positive examples. Since it is assumed that connection statistics follow a multi-variable non-uniform statistical distribution of some kind (i.e. they are regular to a certain extent), the mean and standard deviation of the connection statistics establishes a first definition of the actual sample from which one can infer that variations that violate these distribution could be evaluated as possible cases of identity theft. In an attempt at furnishing a better and more realistic adjustment of the space rendered by the SVM classifier, values of positive examples are interspersed – driven by an interspersing percentage represented by ϑ – over the generated synthetic traces. The value of this percentage must be set so as not to reach over-fitting in the feature space representing the connectivity time behavior of user A, which could eventually cause an increment of false alarms (i.e. connection time samples of the user classified as a suspicious behavior)

3.2 Experimental Results

In order to preliminarily shed light on the performance of our proposed impersonation detection scheme, a set of simulation experiments are next presented. The goal of these simulations is not to characterize the system in terms of their detection performance indicators, but instead to empirically verify that the transformation of the connection traces to the $N = 10$ dimensional alternative feature space listed above balances the trade-off between over- and under-fitting significantly better than the untransformed connection time statistics.

Unfortunately, to the knowledge of the authors no data sets containing connection time statistics for social network users are publicly available. Furthermore, existing interfaces for social networks such as Facebook or Twitter only provide time information about messages rather than usage, which might mask non-interactive activity of the user under consideration and/or an eventual impersonation attacker. As a workaround the connection time traces to be processed have been emulated by resorting to different statistical distributions and surveyed habits of real users, which intuitively reflect actual behaviors of real users. Different Poisson and Gaussian distributions have been utilized for establishing non-uniformly distributed random connections over specific hours stipulated for each user profile: users employing mobile devices with a myriad of short connections per day (e.g. teenagers with more assiduity), users who mainly establish long connections via web interfaces, users that combine both types of connections or users whose account is used as a marquee for advertising his/her business and with a much more regular connection schedule than their counterparts. It is important to note that this approach does not involve any loss of generality for the designed scheme since, as mentioned in the introduction, this step is the first of a complex impersonation detection system which will take into account a broader set of features at the level of social relationships or communities within the platform or the use of language. This first detection phase is specially meaningful if users show some time regularity in their connection habits, being not so decisive otherwise. For the sake of brevity in the foregoing discussion, the analysis focuses on working days due to the more expected connection regularity of the users with respect to the weekend, when the user has more leisure time and their behavioral pattern could be different and irregular. A total of $T^* = 50$ positive examples are fed the classifier as the training set, which correspond to the true connection traces of the user under analysis during 50 days.

Figures 2.a and 2.b represent the two-dimensional reduced projection – via multidimensional scaling with euclidean dissimilarity measure [14] – of the feature space corresponding to user A (in blue) without and with feature extraction, respectively. Correspondingly, blue markers indicate the projected connection time indicators of user A (either original or transformed), whereas red markers correspond to the projections of the synthetic patterns with $\vartheta = 0.5$ as mentioned in Section 3.1. From these plots it is straightforward to conclude that by using the newly derived set of features (Figure 2.b) the transformed space corresponding to user A does not include any suspicious synthetic pattern, as opposed to Figure 2.a where the space contains patterns for potential attackers.

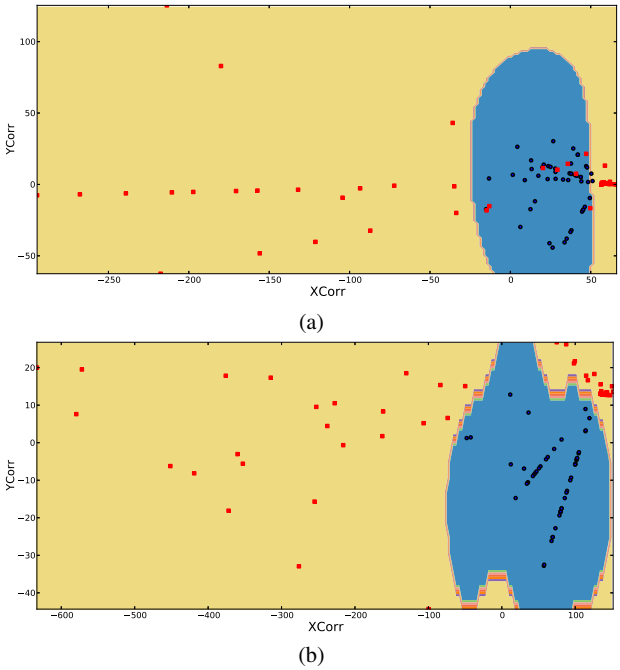


Fig. 2 Two-dimensional representation of the classification space based on original (a) and transformed (b) connection time traces

4 Conclusions and Future Work

This paper has elaborated on a novel approach for detecting impersonation attacks in social networks based exclusively on connection time statistics. The proposed scheme is conceived within a more complex, multi-stage detection system which will address not only the connection schedule and habits of users, but also the generated content (via natural language processing) and the social connectivity of the user, which unveils information about how and with whom he/she interacts. Specifically, this work has focused on creating a behavioral user profile in terms of connection time information, which is deemed an essential feature in social networks as it represents the degree of dependency, availability and regularity of the user at hand. Furthermore, no direct interaction with the social platform is required at this first detection stage, of utmost interest for regulatory limitations on content privacy. This baseline profiling stage provides important information to trigger subsequent alerts regarding a change in behavior that may correspond to a potential impersonation attack. To this end, a SVM classifier has been fed with both this information and a synthetically generated complementary space representing any behavior not previously observed in the user connection history. The inclusion of these synthetic examples allows parametrically forcing the model to fit itself since it is preferable in this scope to trigger many false alarms than to jeopardize the detection capability

of the scheme. Finally, simulation results have verified that by carefully transforming connection information in an alternative feature space a better characterization of the user behavioral profile can be achieved.

Future research will be devoted towards benchmarking this approach jointly with other machine learning schemes such as those belonging to the family of one-class classifiers. This extension will allow verifying whether the generated complementary space (i.e. the synthetic traces for potential impersonation attacks) contributes positively to the classification in terms of its detection indicators.

Acknowledgements. The presented work has been partially supported by the Basque Government under the CYBERSID project grant. The authors would also like to thank Dr. Sergio Gil-Lopez from TECNALIA for fruitful discussions on this research work.

References

1. Pew Internet research on social networking, `http://www.pewinternet.org/Commentary/2012/March/Pew-Internet-Social-Networking-full-detail.aspx` (retrieved on April 2014)
2. Dasgupta, K., Singh, R., Viswanathan, B., Chakraborty, D., Mukherjea, S., Nanavati, A.A., Joshi, A.: Social Ties and Their Relevance to Churn in Mobile Telecom Networks. In: Proceedings of the 11th International Conference on Extending Database Technology, pp. 668–677 (2008)
3. Eysenbach, G.: Medicine 2.0: Social Networking, Collaboration, Participation, Apomediation, and Openness. Journal of Medical Internet Research 10(3) (2008)
4. Zeng, L., Hall, H., Pitts, M.J.: Cultivating a Community of Learners. The Potential Challenges of Social Media in Higher Education. In: Noor Al-Deen, H., Hendricks, J.A. (eds.) Social Media: Usage and Impact. Lexington Books (2011)
5. Twitter: Keeping our users secure, `https://blog.twitter.com/2013/keeping-our-users-secure` (retrieved on April 2014)
6. Martin, A., Anutthamaa, N.B., Sathyavathy, M., Saint Francois, M.M., Venkatesan, P.: A Framework for Predicting Phishing Websites Using Neural Networks. IJCSI International Journal of Computer Science Issues 8(2), 330–336 (2011)
7. Salem, M.B., Stolfo, S.J.: Modeling User Search Behavior for Masquerade Detection. In: Sommer, R., Balzarotti, D., Maier, G. (eds.) RAID 2011. LNCS, vol. 6961, pp. 181–200. Springer, Heidelberg (2011)
8. Egele, M., Stringhini, G., Kruegel, C., Vigna, G.: COMPA: Detecting Compromised Accounts on Social Networks. In: ISOC Network and Distributed System Security Symposium, NDSS (2013)
9. Gao, H., Chen, Y., Lee, K., Palsetia, D., Choudhary, A.: Towards Online Spam Filtering in Social Networks. In: ISOC Network and Distributed System Security Symposium, NDSS (2012)
10. Cortes, C., Vapnik, V.: Support-Vector Networks. Machine Learning 20(3), 273–297 (1995)
11. Miyamoto, D., Hazeyama, H., Kadobayashi, Y.: A Proposal of the AdaBoost-based Detection of Phishing Sites. In: Proceedings of the Joint Workshop on Information Security (2007)
12. Zhang, Y., Hong, J., Cranor, L.: Cantina: A Content-based Approach to Detecting Phishing Web Sites. In: Proceedings of the International World Wide Web Conference, WWW (2007)
13. Liu, W., Huang, G., Liu, X., Zhang, M., Deng, X.: Detection of Phishing Web Pages based on Visual Similarity. In: Proceedings of the International World Wide Web Conference (WWW), pp. 1060–1061 (2005)
14. Borg, I., Groenen, P.: Modern Multidimensional Scaling: Theory and Applications, 2nd edn., pp. 207–212. Springer (2005)

A Study of Machine Learning Techniques for Daily Solar Energy Forecasting Using Numerical Weather Models

Ricardo Aler, Ricardo Martín, José M. Valls, and Inés M. Galván

Abstract. Forecasting solar energy is becoming an important issue in the context of renewable energy sources and Machine Learning Algorithms play an important rule in this field. The prediction of solar energy can be addressed as a time series prediction problem using historical data. Also, solar energy forecasting can be derived from numerical weather prediction models (NWP). Our interest is focused on the latter approach. We focus on the problem of predicting solar energy from NWP computed from GEFS, the Global Ensemble Forecast System, which predicts meteorological variables for points in a grid. In this context, it can be useful to know how prediction accuracy improves depending on the number of grid nodes used as input for the machine learning techniques. However, using the variables from a large number of grid nodes can result in many attributes which might degrade the generalization performance of the learning algorithms. In this paper both issues are studied using data supplied by Kaggle for the State of Oklahoma comparing Support Vector Machines and Gradient Boosted Regression. Also, three different feature selection methods have been tested: Linear Correlation, the ReliefF algorithm and, a new method based on local information analysis.

1 Introduction

Photovoltaic systems are becoming important sources of energy in electricity networks. However, electric utility companies are required to guarantee electricity supply within certain ranges which is difficult given the fluctuating nature of weather conditions. Thus, accurate forecasts of solar radiation is becoming an important issue in the context of renewable energy sources. An approach to forecasting is to use statistical and machine learning techniques based on historical data of solar

Ricardo Aler · Ricardo Martín · José M. Valls · Inés M. Galván
Computer Science Department, Carlos III University, Spain
e-mail: aler@inf.uc3m.es

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_29

production [5]. With respect to the machine learning techniques, many works appear in the literature, that use for instance Artificial Neural Networks [11] or Support Vector Machines [3, 14]. However, for the prediction horizons required by photovoltaic plants (day-ahead), it has been shown that models based on Numerical Weather Prediction (NWP) systems, such as the Global Forecast System (GFS) and the European Centre for Medium-Range Weather Forecast (ECMWF), are a good alternative [5]. These global models predict some meteorological variables for points in a low resolution grid. NWP predicted variables have been used as input for machine learning techniques mainly for wind power prediction [1, 12] and recently for solar energy forecasting [8, 18].

Here, we are interested on the problem of predicting incoming solar energy from NWP models computed from the NOAA/ESRL Global Ensemble Forecast System (GEFS). GEFS provides short-term forecasting for several meteorological variables, for different points or nodes located in a grid. For this paper, we use the data supplied by Kaggle[1] where the goal was to predict the total daily solar energy at 98 Oklahoma solar sites using 15 NWP variables every three hours for a 16×9 grid. In principle the closest grid nodes to the solar station should be the most relevant for prediction, but it can be useful to know how prediction accuracy improves as more and more GEFS grid nodes are used as input for the machine learning techniques. However, using the variables from a large number of grid nodes can result in many attributes which might worsen the generalization capabilities of the learning algorithms. Therefore, our second goal is to study the performance of different feature selection algorithms on prediction accuracy.

The rest of the paper is structured as follows: Sections 2, 3, and 4 describe de data, the regression and the feature selection methods, respectively. Section 5 shows the experimental results including the preliminary studies and parameter adjustment, the study of the influence of the number of grid nodes, and the study of feature selection methods. Finally, section 6 provides the conclusions and future work.

2 Description of Data

The data available from the Kaggle website has been provided by the American Meteorological Society for the 2013-14 Solar Energy Prediction Contest. The goal is to predict the total daily incoming solar energy, measured in $J \times m^2$, at 98 sites of the Oklahoma Mesonet network, which covers a surface of, approximately, 180000 square kilometres. The input data for each day corresponds to the output of the numerical weather prediction model GEFS using 11 ensemble members and 5 forecast timesteps from 12 to 24 hours in 3 hour increments. Each ensemble member produces outputs for 15 different meteorological variables for each timestep and each point of a 16×9 uniform land-surface grid, with a spatial resolution of about 90 km, that includes the State of Oklahoma and surrounding areas. Some of the

[1] `https://www.kaggle.com/c/ams-2014-solar-energy-prediction-contest`

meteorological variables used are the following: accumulated precipitation ($kg.m^{-2}$), air pressure (Pa), downward and upward shortwave/longwave radiation ($W.m^{-2}$), cloud cover ($\%$), temperature (K), etc. A more detailed information can be found in [1]. Thus, the number of attributes for each grid node is $11 \times 5 \times 15 = 875$. Since the number of grid points is $16 \times 9 = 144$, the total amount of available data for each day equals 118800. Data has been collected everyday from 1994 to 2007 (5113 days) in association with the corresponding accumulated incoming solar energy, which is the attribute to be predicted. This accumulated incoming solar energy (in $J \times m^2$) has been calculated by summing the solar energy measured by a pyranometer at each mesonet site every 5 minutes, from the sunrise to 23:55 UTC of the corresponding date.

From the total input-output available data covering 14 years, we have used the period 1997-2005 as the training set (4380 days), reserving the period 2006-2007 (733 days) for the testing set.

3 Regression Methods

Support Vector Machine (SVM) [4] is a class of supervised learning method extensively applied to classification and regression problems. SVMs basically construct maximum margin hyperplanes and use kernel functions to build non-linear models. The Kernel functions most used are linear, polynomial, and the Radial Basis Function (RBF) kernels. Accuracy is greatly influenced by the cost parameter C and the kernel parameters (σ in the case of the most commonly used kernel, the RBF). A more detailed information about SVMs can be found in [2, 16]. In this work, we have used the WEKA SVM implementation called SMO [9].

Gradient Boosted Regression (GBR) is a recent machine learning technique that has shown considerable success in predictive accuracy. The method was proposed by Friedman [6, 7] and it produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. GBR uses two algorithms: regression trees are from the classification and regression tree (decision tree) group of models, and boosting (an adaptive method for combining many simple models to give improved predictive performance) builds and combines a collection of models. Like SVM, the accuracy of GBR models depends on some parameters, as the number of trees used, the shrinkage (a regularization parameter) and the depth of trees. A more detailed description can be found in [13]. We have used for experiments the gbm package [17] from the R language [15].

4 Attribute Selection Methods

In this work, different attribute selection algorithms have been used. They are feature weighting algorithms because they assign weights to input attributes individually,

depending on their relevance to the oputput, and rank them based according to these weights. Two of them are well known algorithms, linear correlation and the ReliefF algorithm [10]. The third one is a new algorithm based on local information analysis, which is described below.

The linear correlation attribute selection method ranks attributes according to their linear correlation with the target. The ReliefF algorithm [10] is also a feature weighting algorithm that estimates the quality of attributes in problems with strong dependencies between attributes. The estimation of the quality of attributes is made according to how well their values distinguish between instances that are near to each other. It evaluates an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and different class. In our work, the algorithm implementation for Weka tool [9] (ReliefFAttributeEval) has been used as the attributes evaluator and Ranker method as search method, which ranks the attributes by their individual evaluations.

Attribute Selection Algorithm Based on Local Information Analysis
The algorithm divides the input space into a grid of fixed-size square regions, called cells. In this work, the algorithm maps all possible subsets of 1 and 2 attributes into grids of dimension dim 1 and 2, respectively. Attributes are ranked according to an evaluation function F_I that measures the information contained in the attributes in each of the cells. Information in a cell is measured as the number of patterns in the cell belonging to class $C1$ that cannot be explained by chance alone, assuming a binomial distribution with parameters (n, p), where n is the total number of patterns in the cell and p is the ratio of $C1$ instances in the whole training set. Thus, given a Cell and a value Conf of the confidence parameter, the information is measured as in Eq. 1 :

$$F_I(\text{Cell}, \text{Conf}) = \max(\text{Cell.C1} - \text{IDF}(\text{Cell.Total}, \text{Conf}, (N_{C1}/N)), 0.0) \quad (1)$$

where Cell.Total = Cell.C0 + Cell.C1; Cell.C1 and Cell.C0 are the number of patterns that belong to $C1$ and $C0$ in the cell, respectively; N_{C1} is the number of $C1$ patterns in the whole training set; N is the number of training patterns; and IDF is the inverse binomial distribution function with parameters n = Cell.Total and $p = N_{C1}/N$. Conf measures the confidence that the distribution of patterns within the cell has been generated by chance. The algorithm uses different values of the confidence parameter from ConfMin to ConfMax with a step $\Delta = 0.25$. The specific algorithm steps are the following:

1. A vector of attribute rankings VR is initialized to zero: $\text{VR}_i = 0 \, \forall i \in \{1, ..., \text{NAtrs}\}$, where NAtrs is the total number of attributes in the problem.
2. An information matrix MI is initialized to zero, for every attribute and confidence level: $\text{MI}_{ij} = 0 \; \forall i \in \{1, ..., NAtrs\}$ and $\forall j \in \{1, ..., C\}$, where $C = (\text{ConfMax} - \text{ConfMin})/\Delta$.

3. Starting with $dim = 2$, a grid of 4^{dim} cells is obtained by dividing the interval $[0, 1]$ in 4 parts. For each combination of dim attributes, the values of attributes are mapped into every cell of the grid.
4. The information provided by each combination of dim attributes in each cell is estimated for each confidence value by using $F_I(\text{Cell}, \text{Conf}_j)$ (Eq. 1), being $\text{Conf}_j = \text{ConfMin} + j * \Delta$ and $i = j, ..., C$. That information is stored in column j of MI.
5. The attribute with highest information for confMax in MI (the last column of matrix MI) is assigned a rank of NAtrs. Next attribute is assigned a rank of NAtrs -1 and so on. This process is continued as long as information is strictly larger than zero. When information is zero, the next confidence value (ConfMax$-\Delta$) is used, and the process is repeated until all attributes have been ranked. The ranking is accumulated in VR.
6. Steps 2 to 5 are repeated for single attributes, i. e. $dim = 1$ (4 cells in the grid).
7. Based on values stored in VR, attributes are ordered. Thus, they are ordered by decreasing relevance.

The algorithm assumes that the output is binary, i.e. patterns belong to two classes, $C0$ and $C1$. For regression, the problem is transformed into 10 binary problems, by discretizing the output value in 10 intervals. The attribute selection algorithm is applied to each problem and the ranking of the 10 set of attributes is combined.

5 Experimental Results

In this work, SMO and GBR have been used to approximate the solar energy production. First, for each solar station, models have been built using the information provided by the 16 nearest grid nodes and then, an attribute selection procedure is carried out. Before running the models, some preliminary studies have been done in order to decide aspects related with the information provided by GEFS and, also, to decide some important parameters of the machine learning algorithms.

5.1 Preliminary Studies and Parameter Adjustment

As it has been mentioned in section 2, data provided by GEFS includes 11 ensemble output forecasting models. Using the 11 ensemble members as input variables to ML algorithms would imply to build up 11 regressors for each mesonet station. On the other hand, it is not obvious which ensemble member should be chosen. In this work, three different approaches to combine the 11 ensemble members have been considered: compute the mean of the 11 ensembles, compute the median, and compute the mode. The three approaches have been run using the information provided by the 5 nearest grid points and the average of MAE (mean absolute error) for

the 98 mesonet stations are 1940816, 1955128 and, 1979554, respectively. Therefore, we have decided to use the mean of the 11 ensemble models to summarize the information provided by all the ensembles.

On the other hand, the accuracy of SMO and GBR models depends highly of their parameters. To establish the optimum parameter values for each mesonet and for each possible number of grid nodes would involve a very heavy computation. Then, the parameters of models have been selected using only the first of the 98 mesonet (ACME station) and the five nearest grid nodes. A two-year validation dataset has been used to compare the different parameter combinations. An exhaustive grid search has been run to locate the optimal parameters (the cost parameter C and σ for SMO, and number of trees, shrinkage, and tree depth for GBR). Experiments established that for SMO with linear kernel (linear-SMO), the best parameter is $C = 0.03$. For SMO with RBF kernel (RBF-SMO) the best parameters are $C = 1$ and $G = 0.01$. For GBR models, they are: number of trees=5000, shrinkage=0.01, and tree depth=10. Those parameters have been used for all the experiments in the next sections.

5.2 *Prediction Accuracy with Respect to the Number of GEFS Grid Nodes*

Figure 1 displays the evolution of MAE as the number of GEFS grid points is increased from 1 to 16 for linear-SMO, RBF-SMO, and GBR. Averaged train and test MAEs for 98 solar sites are shown on the left and right figures of 1, respectively.

With respect to test MAE, it can be seen that the two non-linear models GBR and RBF-SMO perform significantly better than the linear one (linear-SMO). In all cases, it is observed that MAE tends to improve as the number of grid points increases. Both RBF-SMO and GBR obtain similar results when the number of grid points is large (8 or more). But GBR performs better when only a few grid points are used (from

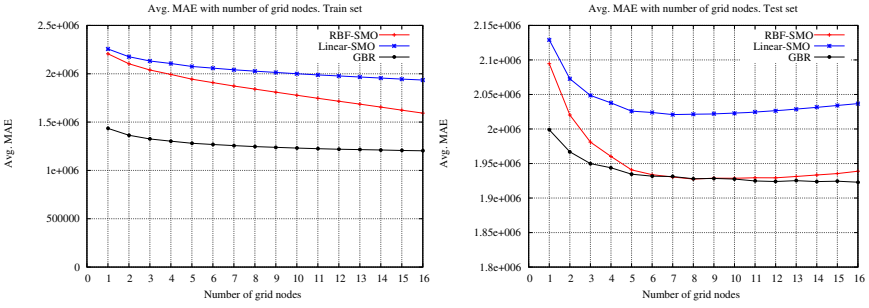


Fig. 1 Average MAE for different number of grid nodes, using linear-SMO, RBF-SMO, and GBR. Training and testing set

1 to 4) and does not suffer from the slight overfitting observed for SMO for more than 10/11 GEFS grid points.

The main conclusions from this study are that non-linear models perform much better than the linear one, and that interestingly, the best results are obtained using more than the closest four or five grid nodes (i.e. the grid nodes surrounding the station): the minimum error is obtained from 8 grid points for RBF-SMO and from 16 points for GBR (although the gain obtained by GBR from 8 to 16 points is very small: a 0.26% decrease).

5.3 Study of Feature Selection Methods

Here, the three feature selection algorithms have been applied to all the features present in 16 grid points (16*75=1200 features). The 1200 attributes are ranked and both RBF-SMO and GBR algorithms are trained and tested using the first 400, 500, 600, 800, 900, 1000 attributes, respectively. Figures 2 and 3 display the average MAE for training and testing for the 98 stations obtained using the different numbers of attributes.

Results show that, surprisingly, although the original number of attributes is very large, the different attribute selection methods do not improve prediction error in general. Therefore, in this domain all 1200 attributes seem relevant to some degree. However, results also show that the number of attributes can be greatly reduced without loosing a significant accuracy. In the case of RBF-SMO, the local information analysis algorithm allows to reduce the number of attributes from 1200 to 600 and obtain the same error (1938241 with 600 features vs. 1938855 with all features). In this case, ReliefF and linear correlation obtain higher errors for the same number of (600) attributes (1965442 and 1997274, respectively). When GBR is used as regressor, the number of features cannot be reduced to the same extent but with 800 features, ReliefF and the local information algorithms are able to

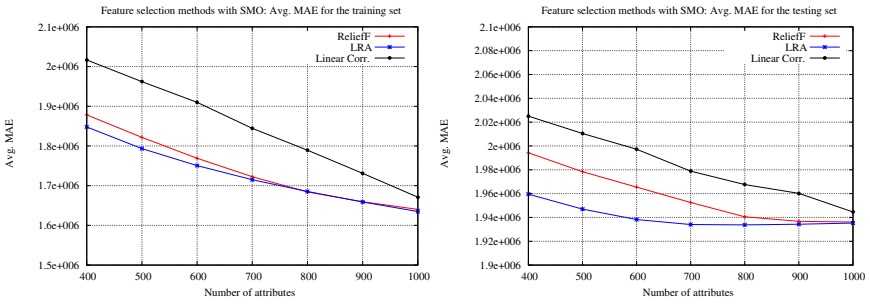


Fig. 2 Average MAE for different number of attributes, using RBF-SMO. Training and testing set

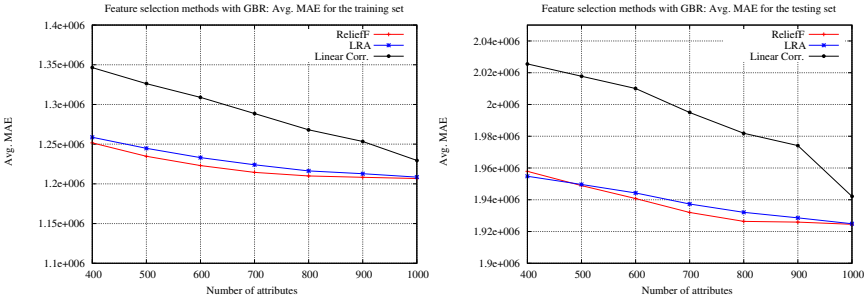


Fig. 3 Average MAE for different number of attributes, using GBR. Training and testing set

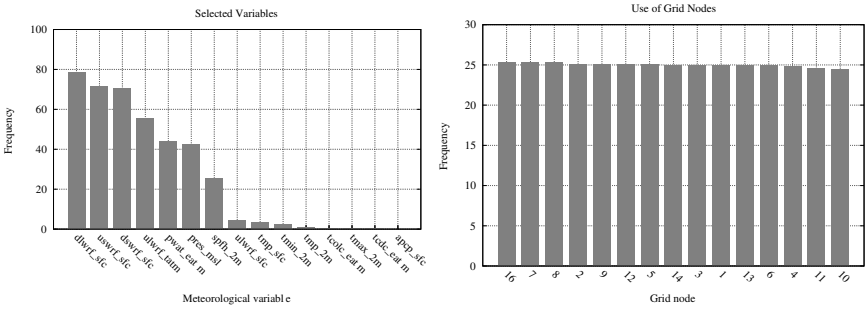


Fig. 4 (a) Bar graph of Meteorological Variables. (b) Bar graph of GEFS grid nodes

obtain quite similar errors compared to the full set of features (1926369 and 1932069, respectively, versus 1922594 for the 1200 features). In all cases linear correlation is not competitive with the other methods.

Finally, we will take advantage of the attribute ranking performed by the attribute selection methods in order to know which are the most relevant meteorological variables and the most relevant grid points. For this purpose, we have used the local information algorithm to select the 400 most relevant features. Figure 4 displays bar graphs of the variable names and the grid points used, from 1 (the closest) to 16, respectively. Figure 4 (a) shows a clear preference for some of the variables (downward long-wave radiative flux average at the surface, upward short-wave radiation at the surface, downward short-wave radiative flux average at the surface, and upward long-wave radiation at the top of the atmosphere, . . .). However, the flatness of graph 4 (b) shows no preference for closer vs. farther away grid nodes: all grid points have about the same amount of attributes present in the 400 most relevant attributes.

6 Conclusions

In this work, we have performed an study of different machine learning techniques in the context of solar energy forecasting using NWP models computed from the NOAA/ESRL Global Ensemble Forecast System (GEFS) for different nodes located in a grid. On one hand, three different regression methods (linear SVM, RBF-SVM, and GBR) have been used to build forecasting models and to study the influence of the grid nodes number on prediction accuracy. On the other hand, given the large number of features in this domain, three different attribute selection methods have been tested (linear correlation, ReliefF, and a local information analysis algorithm).

Experimental results show that the non-linear methods obtain lower errors than the linear one. GBR and RBF-SMO perform similarly, although RBF-SMO shows some slight overfitting when the number of grid points is large. Also, in the case of the best performing method (GBR), forecasting accuracy tends to improve as the number of GEFS grid nodes used as input increases, even beyond the 4 or 5 closest nodes. Contrary to what was expected, feature selection was not able to improve solar energy prediction, although with RBF-SMO, the local information algorithm can obtain similar predictions with a half of the attributes.

In the future, it would be interesting to extend this study to other situations where geographical or meteorological features are different (surface elevations, different pressure levels grid nodes) or to other prediction problems within the renewable energy domain involving grid numerical weather prediction models.

Acknowledgements. The authors acknowledge financial support granted by the Spanish Ministry of Science under contract TIN2011-28336(MOVES).

References

1. Alaíz, C.M., Torres, A., Dorronsoro, J.R.: Sparse linear wind farm energy forecast. In: ICANN 2012, Part II. LNCS, vol. 7553, pp. 557–564. Springer, Heidelberg (2012)
2. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2(2), 121–167 (1998)
3. Chen, J.-L., Liu, H.-B., Wu, W., Xie, D.-T.: Estimation of monthly solar radiation from measured temperatures using support vector machines–a case study. Renewable Energy 36(1), 413–420 (2011)
4. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3), 273–297 (1995)
5. Diagne, M., David, M., Lauret, P., Boland, J., Schmutz, N.: Review of solar irradiance forecasting methods and a proposition for small-scale insular grids. Renewable and Sustainable Energy Reviews 27, 65–76 (2013)
6. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of Statistics, 1189–1232 (2001)
7. Friedman, J.H.: Stochastic gradient boosting. Computational Statistics & Data Analysis 38(4), 367–378 (2002)
8. Gala, Y., Fernández, A., Dorronsoro, J.R.: Machine learning prediction of global photovoltaic energy in spain. In: International Conference on Renewable Energies and Power Quality, number 12 (2014)

9. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. ACM SIGKDD Explorations Newsletter 11(1), 10–18 (2009)
10. Kononenko, I.: Estimating attributes: analysis and extensions of relief. In: Bergadano, F., De Raedt, L. (eds.) ECML 1994. LNCS, vol. 784, pp. 171–182. Springer, Heidelberg (1994)
11. Mellit, A., Pavan, A.M.: A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected {PV} plant at trieste, Italy. Solar Energy 84(5), 807–821 (2010)
12. Monteiro, C., Bessa, R., Miranda, V., Botterud, A., Wang, J., Conzelmann, G., et al.: Wind power forecasting: state-of-the-art 2009. Technical report, Argonne National Laboratory, ANL (2009)
13. Schonlau, M.: Boosted regression (boosting): An introductory tutorial and a stata plugin. Stata Journal 5(3), 330 (2005)
14. Sharma, N., Sharma, P., Irwin, D., Shenoy, P.: Predicting solar generation from weather forecasts using machine learning. In: 2011 IEEE International Conference on Smart Grid Communications (SmartGridComm), pp. 528–533. IEEE (2011)
15. R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2014)
16. Vapnik, V.N.: Statistical learning theory (adaptive and learning systems for signal processing, communications and control series). John Wiley & Sons, A Wiley-Interscience Publication, New York (1998)
17. Greg Ridgeway with contributions from others. gbm: Generalized Boosted Regression Models. R package version 2.1. (2013)
18. Wolff, B., Lorenz, E., Kramer, O.: Statistical learning for short-term photovoltaic power predictions. In: DARE: Data Analytics for Renewable Energy Integration. European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (2013)

AGGE: A Novel Method to Automatically Generate Rule Induction Classifiers Using Grammatical Evolution

Romaissaa Mazouni and Abdellatif Rahmoun

Abstract. One of the main and fundamental tasks of data mining is the automatic induction of classification rules from a set of examples and observations. A variety of methods performing this task have been proposed in the recent literature. Many comparative studies have been carried out in this field. However, the main common feature between these methods is that they are designed manually. In the meanwhile, there have been some successful attempts to automatically design such methods using Grammar-based Genetic Programming (GGP). In this paper, we propose a different system called Automatic Grammar Genetic Programming (AGGP) that can evolve complete java program codes. These codes represent a rule induction algorithm that uses a grammar evolution technique that governs a Backus Naur Form grammar definition mapping to a program. To perform this task, we will use binary strings as inputs to the mapper along with the Backus Naur Form grammar. Such binary strings represent possible potential solutions resulting from the initialized component and Weka building blocks, this would ease the induction process and makes induced programs short. Experimental results prove the efficiency of the proposed method. It is also shown that, compared to some recent and similar manual techniques (Prism, Ripper, Ridor, OneRule) the proposed method outperforms such techniques.A benchmark of well-known data sets is used for the sake of comparison.

Keywords: AGGE: Automatic Generation of classifiers using Grammatical Evolution, Grammatical Evolution, Context Free Grammar, Rule Induction Algorithms, Data mining, Rule Based Classification.

1 Introduction

The use of Evolutionary Algorithms (EA) in Artificial life that simulates the natural process of evolution started in the 1960's with the works of Alex Fraser and

Romaissaa Mazouni · Abdellatif Rahmoun
Djillali Liabes University, Computer Science Department, Algeria
e-mail: rom.mazouni@yahoo.com, Rahmoun_abd@yahoo.fr

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_30

Nill Bricelli, the writings of John Holland on Genetic Algorithms and Ingo Rechenberg on evolution strategies [1]. Koza [2] made this field very popular. The dramatic increases in the power of computers allowed the researchers to solve complete complicated real world problems inducing the automatic generation of computer programs using bio-inspired techniques such as evolutionary algorithms which have proven their ability to solve multidimensional problems more efficiently than software produced by human designers. Genetic Programming (GP) is the most popular EA technique, to automatic generation of computer programs using genetic operators. However in their standard form GP algorithms have some issues that need to be taken in consideration while designing it.One of the major problems faced when designing the GP is the selection of the function set and the terminal set.These sets should be chosen so as to satisfy the closure property [3]. The closure property requires that each function in the function set should be able to handle and process the values received as inputs to it, either these inputs are terminals or outputs of other functions. In order to respect this property the early designed GP algorithms dealt with only one data type which reduced of course the ability and power of GP. Recent GP systems use new approaches to give the GP the power to handle different data types,these approaches were used to satisfy the closure property, Grammar based genetic programming [4] is the technique the mostly used to tackle the closure problem. Another problem faced when using genetic programming systems in the huge search space that the GP will have to scan, using grammars along with GP was a solution to this issue because grammars offers the ability to introduce prior knowledge about the solutions basic structure to the GP so it will be easy to restrict the search space. Insuring the syntactic validity of the individuals is also a shutter that we should think about when using the GP , Grammars are also a way to assure the solutions syntactic validity and to also preserve this validity after applying genetic operators.

One of the most popular research fields in computer science beside the evolutionary algorithms is datamining and more specifically the classification area, there exist several approaches to perform classification, and the most used technique among these latters is the rule based classification. Rule based classifiers are frequently used due to the humanly comprehensible nature of the models they generate. These formers have evolved through the time, from using simple sequential covering concepts in simple algorithms such as PRISM and CN2 to algorithms using new concepts such as minimum description length in the RIPPER algorithm ,these old and novel components can be modified and gathered automatically in different ways to produce new algorithms.

2 Grammar Based Genetic Programming

Writing a computer program is a purely manual task that is time consuming and requires a lot of reflection, it is a really tedious task. The idea of automating this process appeared in the 1950's when Arthur Samuel thought of giving computers

the ability to learn without being explicitly programmed. This process was called back then Machine Learning, the in the 1980's the Machine Learning definition was changed by Tom Mitchell into computers having the ability to learn via experience , at the same time another computer science field has appeared which was directly applied to automatic code generation this latter was called Genetic Programming. After several years of research in this new field a new branch of it called Grammar Based GP appeared and was conceived in order to satisfy the closure property and to restrict the search space as well.

2.1 Context Free Grammar Based Genetic Programming (CFG-GP)

The use of formal grammars was firstly introduced by Peter Whigham in order to control the search algorithm of genetic programming, it was also a solution provided to solve the typing problem recognized by [9] and a mean to introduce more bias into the genetic programming. Peter Whigham proposed a method called Context Free Grammar GP, [4, 10] and noted that the use of CFG can be in a similar way to that of typing that restricts the structure of candidate solutions , this new method is based on the redefinition of the elements of tree based GP so it respects a certain grammar G.Individuals in CFG-GP have the tree structure and they are derived according to the CFG introduced to the GP, the genetic operators are modified in a way to preserve this representation.

Context Free Grammars allow to easily use programming languages most appropriately for a given problem, [12].

2.2 Grammatical Evolution (GE)

Grammatical evolution is a special case of grammar based genetic programming that uses a mapping procedure between the genotypes and phenotypes of individuals [6, 13]. When using GE to evolve solutions of a certain problem we don't need to give attention to the operators and how they will be implemented, the GE brings the benefit of the validity of the programs being evolved for free. The grammatical evolution described by [17] marries principles from molecular biology to the representational power of formal grammars [4]. The genotype-phenotype mapping in GE allows the operators to act on genotypes and not solution trees as in traditional GP and this is what makes this technique attractive.In analogy to the biological DNA the string of integers that represent the genotype is called Chromosome and the values it is consisted of are called Codons, A BNF grammar definition must initially be introduced when using GE to solve a certain problem, this BNF grammar describes the output language produce d by the system in [19] and it is used along with the chromosomes in the mapping process. The mapping process is the process of

mapping non-terminals to terminals and that is done by converting the binary string data structure into a string of integers, which is brought from the Genetic Algorithm into the machinery of GE. Then this integers string is passed through a translation process, where the rules in the BNF definition are selected. The production rules, equivalent to amino acids in genetics, combine to produce terminals, which are the components making up the final program. One problem that could be faced when mapping binary strings is short genes (when we run out of genes but we still have non-terminals to map), a solution to this issue is wrapping the individual and reuse the genes, a gene in GE could be used several times in a mapping process; We can also declare the individual invalid and punish it with a suitably harsh fitness value. The rules selection is performed using the modulo operator and each time we read a codon (an integer) we divide it by the number of the rule's choices and the remainder of this division is the number of the rule to be selected.

3 Sequential Covering

Rule induction is an area of data mining where formal rules are extracted using a certain dataset, these rules will represent local patterns of a full scientific model in this dataset. One of the paradigms of the rule induction process is decision rules, which is used to induce decision rules using a certain set of observations and it has two different methods to do it, the first method is the indirect one where we extract the decision rules from other knowledge representations and the second method is the direct one, when using this method we extract decision rules directly from the training set.

In this paper we will focus on the second method (direct one). Sequential covering is a technique following this paradigm. The idea of this technique is to learn one rule, remove the examples it covers and then, repeat this process [15] as described in the following pseudo code.

The Sequential Covering Pseudo Code

```
SequentialCovering(target_att,atts,examples,threshold)
    Learn_rule ={ };
    Rule=Learn_One_Rule(target_att, atts ,examples);
    While ( performace(rule, examples) > threshold ) do
           Learned_rules=learned_rules +rule;
           examples= example - examples correctly classified by rule;
           rule= Learn_One_Rule( target_att,atts,examples);
   Done
   Learned_rules = sort(Learned_rules(performance));
Return Learned_rules;
```

In the sequential covering algorithms, Learn_One_Rule should have high accuracy but not necessarily high coverage, and it is not guaranteed to find the best or the smallest set of rules because this method performs a greedy search.There are plenty of proposed algorithms that follow the sequential covering paradigm. These algorithms differ in 4 main components of the sequential covering that can differ in the

way they represent the candidate rules. Some use the propositional logic such as: CN2 and Ripper, others use the first order logic such as: FOIL and REP. The algorithms following sequential covering can also use different search mechanisms [16], there are three different search strategies, the bottom-up one where we start with a random example of the dataset then we generalize it, the second strategy we have is top-down one where we start with an empty rule then we specialize it by adding preconditions to it and finally the bidirectional one. There exist also two different search methods the most used one is the greedy method (ex: PRISM) and the beam method (ex: CN2, BEXA). Covering algorithms have different manners to evaluate rules some of the existing methods are : the confidence (ex: SWAP -1), the Laplace estimation (ex: BEXA,CN2), the m-estimate, the ls-content (ex : HYDRA), the minimum description length and the info gain. The final component that differentiate the covering algorithms is the pruning methods, pruning is used to handle over fitting and noisy data , there exist two kinds of pruning : the pre-pruning that deals with the over fitting and noisy data and the post-pruning that deals with rejection and conflict problems in order to find a complete consistent rule set. Pre-pruning gives the ability to find a fast model while the post-pruning helps finding simpler and more accurate models.

Pappa and al [16], proposed a full Context Free Grammar using a Backus Naur Form terminology that presents all elements necessary for building a basic rule based classifier following the sequential covering method , this grammar contains 26 production rules, each one representing Non-Terminal symbols and 83 Terminal symbols describing these elements. This grammar produces either a rule list where rules are executed in a certain order or a rule set when there is no order needed when applying rules. This gives different initialization, refinement and evaluation techniques, the grammar is presented in Figure.1.

4 Automatic Generation of New Rule Based Classifiers

The Context Free Grammar based Genetic Programming has been used in order to induce rule based classifiers by [16] and it is to the best of our knowledge the first and only method used to automatically design a rule induction algorithm. It would be very interesting to try to design another system or method that will perform the same task but with less effort spent while designing the system, Grammatical evolution in an interesting method because we do not need to modify the crossover neither the mutation so they can respect the grammar. Grammatical Evolution is the proposed method that we will try to use to automatically generate rule based classifiers. there exist plenty of classification methods such: Support Vector Machine, Bayesian Neural Networks, Artificial Neural Networks, Decision Trees,...etc. The decision rules model is chosen because it has the tendency to be intuitively comprehensive by human beings. We will propose in the following section a system combining grammatical evolution with a context free grammar to evolve code fragments having the ability to generate accurate, noise tolerant and compact decision rule sets.

4.1 Proposed System

The proposed system has five main components. Initially all we need a grammar that will represent the overall structure of all rule based classifiers, following the sequential covering paradigm that are manually designed. Secondly we need some building blocks taken from Weka to facilitate the task of reading arff files and testing the newly generated classifiers, this can be seen as " code reuse". we need also some of machine learning data sets to train and test these classifiers , we have downloaded these data sets from the UCI machine learning repository [18]. We need multiple datasets so that when we train the rule based classifiers (candidate solutions) they won't be tailored to a certain specific domain. Finally, we have the mapper of the GE [12] that we modified in a way that when it reads terminals, it inserts java code representing this terminal's actions, we used the GEVA [19] frame work to implement the system.

```
<Start>              ::= ( <CreateRuleSet> | <CreateRuleList> )  [<PostProcess>]
<CreateRuleSet>      ::= forEachClass <whileLoop> endFor <RuleSetTest>
<CreateRuleList>     ::= <whileLoop> <RuleListTest>
<whileLoop>          ::= while <condwhile> <CreateOneRule> endWhile
<condwhile>          ::= uncoveredNotEmpty | uncoveredGreater (10| 20|  90%|  95%|  97%|  99%)  trainEx
<RuleSetTest>        ::= lsContent | confidenceLaplace
<RuleListTest>       ::= appendRule | prependRule
<CreateOneRule>      ::= <InitializeRule> <innerWhile> [<PrePruneRule>] [<RuleStoppingCriterion>]
<InitializeRule>     ::= emptyRule | randomExample|  typicalExample| <MakeFirstRule>
<MakeFirstRule>      ::= NumCond1| NumCond2| NumCond3| NumCond4
<innerwhile>         ::= while (candNotEmpty|  negNotCovered) <FindRule> endWhile
<FindRule>           ::= <RefineRule> <EvaluateRule> [<StoppingCriterion>] <SelectCandidateRules> |
                         <innerIf> <EvaluateRule> [<StoppingCriterion>] <SelectCandidateRules>
<innerIf>            ::= if <condIf> then <RefineRule> else <RefineRule>
<condIf>             ::=  <condIfExamples> | <condIfRule>
<condIfRule>         ::= ruleSizeSmaller (2| 3| 5| 7)
<condIfExamples>     ::= numCovExp (  > |  <)(90%| 95%|  99%)
<RefineRule>         ::= <AddCond>|  <RemoveCond>
<AddCond>            ::= Add1| Add2
<RemoveCond>         ::= Remove1| Remove2
<EvaluateRule>       ::= confidence | Laplace | infoContent|  infoGain
<StoppingCriterion>  ::= MinAccuracy (0.6| 0.7| 0.8)| SignificanceTest (0.1|  0.05| 0.025| 0.01)
<SelectCandidateRules> ::= 1CR| 2CR|  3CR|  4CR|  5CR|  8CR|  10CR
<PrePruneRule>       ::= (1Cond|  LastCond| FinalSeqCond) <EvaluateRule>
<RuleStoppingCriterion> ::= accuracyStop (0.5|  0.6| 0.7)
<postProcess>        ::= RemoveRule EvaluateModel| <RemoveCondRule>
<RemoveCondRule>     ::= (1Cond|  2Cond| FinalSeq) <EvaluateRule>
```

Fig. 1 The Grammar Describing Sequential Covering Method Elements

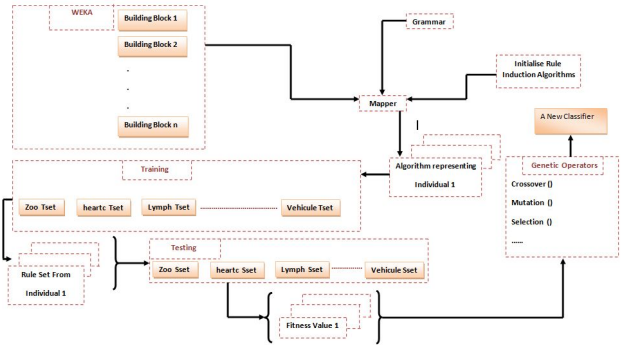


Fig. 2 Modules of The Proposed System

The most important module in the system is the mapper, which needs to be able to read the integer values from the chromosomes (candidate solutions which will be called in this paper AGGE-classifiers) then choose the appropriate corresponding rule of a certain non-terminal and import some of the already coded weka building blocks an insert it along with the terminals corresponding java code into the buildClassifier class of the new AGGE classifier, Figure 6 is a schema representing steps of the global system.

Individuals in this system are represented as integer arrays that will be mapped into rule based classifiers, each array will be read integer by integer. Integers will be divided by number of choices of the current rule. The remainder on the division will be used by the mapper to choose the next rule to be applied. The first integer of each individual will be divided by 2 (the number of choices of the first rule Start) and according to the remainder we will be constructing either a rule set (integer MOD 2 = 1) or a rule list (integer MOD 2 = 0).

When using evolution to solve any problem we need a measure so to enable selecting best individuals among a population of solution, the metric or the fitness function used in this work to evaluate AGGE-classifiers generated during the evolution process is the accuracy method. After the initialization of the first population, individuals (AGGE-classifiers) will undergo the mapping process then they will be represented in java programs, these java programs are actual classifiers that will be compiled then executed (trained and tested) using different data sets and each classifier will have a set of different accuracies (an accuracy for each data set). these accuracies will be then averaged and used as the classifier's overall accuracy so we will be able to compare these AGGE-classifiers in a population. The following equation defines the overall accuracy of an AGGE-classifier i in a given population where $acc_{i,j}$ is the accuracy of this AGGE-classifier i using the data set j and h is the number of data sets.

$$f_i = (\sum_{j=0}^{h} acc_{i,j})/h$$

When using Grammatical Evolution we don't need to think about the consistency with the grammar of new offsprings generated after a mutation or a crossover operation because these latters are applied on the genotypes.

5 Experimentation Results and Discussion

Prior to testing the system we downloaded data sets that we used to train and test the system, we downloaded 19 data sets from [18], these data sets are from different public domains so that the system will not be tailored to a specific domain as mentioned earlier. Some of these data sets have only nominal attributes, others have only numerical attributes, and some data sets have both attributes types. The data sets were divided into 2 groups, 70% of them (13) were used to train then to validate the models, and the rest were used for the purpose of testing. We should note that

this division was done randomly. The meta-training set contains(Monks-2, Monks-3, Balance-scale, lymph, promoters, splice, vowel, vehicle, pima, glass,sonar, hepatitis, ionosphere) and the meta-testing set contains (Monks-1, segment, crx, sonar, heart-c, mushrooms), before starting the training phase each set of the meta-training set is subsampled using the 5-fold cross-validation in order to avoid overfitting and to make predictions more generalizable. The system needs 3 components to start the evolution process. The grammar mentioned earlier in section 3, the meta data sets and finally the grammatical evolution parameters, the number of generation was set to 40, the population size to 200, the mutation probability was set to 0.01, the crossover probability to 0.7, the selection method chosen is the tournament selection and we used the generational replacement. We should mention here that these parameters are not optimized but rather empirically chosen after analyzing a certain number of trials. In order to evaluate the newly generated classifiers we computed the accuracies of 4 manually designed rule based classifiers using all 19 datasets. The first column of Table 1 reports accuracies of the new generated classifier (AGGE-classifier) while the remaining columns shows accuracy values of the 4 manually designed classifiers (Ripper, Ridor, OneRule and Prism), the first 13 rows reports the accuracies of the rules sets generated by the AGGE-classifier the 4 baseline. Using only the meta-training set(each row represent the test accuracy of a single set from the meta training set) these accuracy values are reported here to show the success of the training phase while the last six rows of the table shows the real predictive accuracy values because the AGGE-classifier has never met these sets during the training or validation phase.

The aim of this work was to propose a system that uses the grammatical evolution method to automatically generate rule based classifiers having an accuracy that

Table 1 Accuracy rates (%) Using Both Meta-Sets

	AGGE Classifier	Prism	Ripper	Ridor	OneRule
monks-2	53.8462	37.8698	53.2544	50.8876	42.0118
monks-3	36.8852	30.3279	45.9016	46.7213	37.7049
promoters	90.5660	66.0377	78.3019	74.5283	69.8113
splice	99.1223	70.1823	93.6991	92.1003	24.3574
vowel	82.2222	83.0303	69.6970	77.6768	31.8182
vehicle	64.4526	66.7849	68.5579	70.5674	51.4184
pima	75.3404	70.0349	75.1302	75.0000	70.1823
glass	62.6923	57.4939	66.8224	64.0187	58.4112
zoo	100.000	62.3762	86.1386	940594	42.5743
lymph	68.9189	75.6757	77.7072	85.1351	85.1351
balence scale desc	78.7225	52.3200	80.0800	79.5200	56.3200
ionosphere	88.1474	91.1681	89.7436	88.0342	80.9117
hepatites	68.3871	78.0645	78.0645	78.7097	83.2258
segment	94.2857	92.2944	95.4978	96.1472	64.8052
crx	91.4493	77.5362	85.5072	83.3333	85.7971
mushrooms	100.000	100.000	100.000	100.000	98.5229
monks-1	49.236	26.6129	49.1995	51.6129	50.0000
sonar	76.4423	74.0385	73.0769	73.5577	62.5000
heart-c	80.5291	76.8977	81.5182	79.538	71.6172

can be at least competitive with the existing manually designed classifiers. After the implementation and the testing phase, the system proved its ability to produce classifiers that are highly competitive with the human designed ones. We can easily notice the performance of these formers in Table 1, we should note that we used a 5-fold-cross-validation so we can be able to train the classifiers with the complete data set and then indirectly to test its with all data, this helps to take advantage of all knowledge available in the data sets and to obtain reliable performance results. We should also mention that accuracies in Table 1 were obtained by averaging the accuracy of the rule model generated by the AGGE-classifier for each test set over the 5 iterations of the 5-fold-cross-validation method used during experiments, this also applies on the rest of the benchmark classifiers used for the purpose of comparison. It is worth mentioning that in Table 1 the new generated classifier has practically the same results as the other methods and if we compare only the baseline methods with eah others we can clearly notice that the RIPPER records 10 wins over 5 for PRISM and RIDOR and 2 for the OneRule algorithm and this due to the sophisticated nature of the RIPPER classifier. It uses a growing, pruning and and optimization phase and the Minimum Description Length (MDL) method as a stopping criterion when constructing the rules. Now if we look at the AGGE-classifier accuracies we can notice how close are these accuracies to the baseline algorithms accuracies which is very interesting due to the fact that the AGGE-classifiers is automatically generated, and this removes a great deal of necessary time doing coding tasks. Humans designers can easily go wrong when parameterizing an algorithm during the design process, on the other hand the chance of having bad parameters when using automatic evolution of algorithms is very low. The last six rows show that the AGGE-classifier records 3 wins against the baseline classifiers (crx,Mushrooms,sona), for the heart-c and segment data sets the AGGE results where very close the best accuracy (80.5291 versus 81.5182 and 94.2857 versus 95.4978) these results proved that the proposed approach can be very interesting. However, the AGGE system is time consuming while evolving AGGE-classifiers and requires high computational power. Moreover, it can't be run properly under an ordinary computer, where the evolution process can take up to one week of continuous calculation, we should also note that this version of the system does not handle missing values and thus requires eliminating instances with missing value before using the datasets. Concerning numeric values the system uses the discretization method.

6 Conclusion

This paper proved the possibility of generating automatically rule based classifiers using grammar evolution.The automatically and genetically evolved classifiers can produce results that are competitive with those produced using manually designed algorithm, Results obtained using the newly generated rule based classifier proved the efficiency and effectiveness of this approach, however there are still some gaps to be tackled.

One of the interesting research directions that should be considered to improve the system is to focus on the fitness. This task can be fulfilled by using multi objective fitness function that considers either rules and rule set size or time consumption when performing classification (for real-time classification applications) along with the accuracy. This method might help in finding more efficient and powerful algorithms. Another way for improving the system may be the extension of the grammar so it can produce more complex algorithm structure,through the utilization of the fuzzy rule representation. Subsequently, the system will be able to produce fuzzy rules based classifiers, are more expressive and more powerful than classical ones.

References

1. Beyer, H.G., Schwefel, H.P.: Evolution strategies: A comprehensive introduction. Natural Computing 1, 3–52 (2002)
2. Koza, J.: Genetic programming: on the programming of computers by means of natural selection. MIT Press (1992)
3. Pappa, G.L., Freitas, A.: Automating the Design of Data Mining Algorithms. Springer, Heidelberg (2010)
4. McKay, R., Hoai, N.X., Whigham, P.A., Shan, Y., O'Neill, M.: Grammar-based Genetic Programming: A review. Genetic Programming and Evolvable Machines, 365–396 (2010)
5. Wong, M.L., Leung, K.S.: Applying logic grammars to induce sub-functions in genetic programming. In: IEEE International Conference on Evolutionary Computation, vol. 2, pp. 737–740 (1995)
6. O'Neill, M., Hemberg, E., Gilligan, C., Bartley, E., McDermott, J., Brabazon, A.: GEVA: Grammatical Evolution in Java. SIGEVOlution ACM 3, 17–23 (2008)
7. Norman, P.: Genetic programming with context-sensitive grammars. Phd thesis, Saint Andrew's University (2002)
8. Wong, M.L., Leung, K.S.: Data Mining Using Grammar-Based Genetic Programming and Applications. Kluwer Academic Publishers (2000)
9. Montana, D.J.: Strongly typed genetic programming. Evolutionary Computation Journal 3, 199–230 (1995)
10. McKay, R., Hoai, N.X., Whigham, P.A., Shan, Y., O'Neill, M.: Grammar-based Genetic Programming: A review. Genetic Programming and Evolvable Machines 11, 365–395 (2010)
11. McKay, R.I., Nguyen, X.H., Whigham, P.A., Shan, Y.: Grammars in Genetic Programming: A Brief Review. In: Progress in Intelligence Computation and Intelligence: Proceedings of the International Symposium on Intelligence, Computation and Applications, pp. 3–18 (2005)
12. Nohejl, A.: Grammar Based Genetic Programming. MSc Thesis, Charles University of Prague (2011)
13. Nohejl, A.: Grammatical Evolution. BSc Thesis, Charles University of Prague (2009)
14. Freitas, A.A.: Data mining and Knowledge Discovery with evolutionary algorithms. Springer (2002)
15. Bing, L.: Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Springer Edition (2011)
16. Pappa, L.G.: Automatically Evolving Rule Induction Algorithms using Grammar-based Genetic Programming. Phd Thesis, Kent University (2007)
17. Dempsey, I., O'Neill, M., Brabazon, A.: Foundations in Grammatical Evolution for Dynamic Environments. SCI, vol. 194. Springer, Heidelberg (2009)
18. UCI Machine Learning Repository, `http://archive.ics.uci.edu/ml/`
19. Oltean, M., Grosan, C.: A Comparison of Several Linear Genetic Programming Techniques. Complex Systems Journal 14, 285–313 (2004)

Part IX
P2P, Self-organized and Ubiquitous Systems

Expansion Quality of Epidemic Protocols

Pasu Poonpakdee and Giuseppe Di Fatta

Abstract. Epidemic protocols are a bio-inspired communication and computation paradigm for large and extreme-scale networked systems. This work investigates the expansion property of the network overlay topologies induced by epidemic protocols. An expansion quality index for overlay topologies is proposed and adopted for the design of epidemic membership protocols. A novel protocol is proposed, which explicitly aims at improving the expansion quality of the overlay topologies. The proposed protocol is tested with a global aggregation task and compared to other membership protocols. The analysis by means of simulations indicates that the expansion quality directly relates to the speed of dissemination and convergence of epidemic protocols and can be effectively used to design better protocols.

Keywords: epidemic protocols, expander graphs, extreme-scale computing, decentralised algorithms.

1 Introduction

In extreme-scale distributed systems, computing and spreading global information is a particularly challenging task. Centralized paradigms are not suitable, as they introduce bottlenecks and failure intolerance; fully decentralised and fault-tolerant approached are very desirable.

Epidemic (or Gossip-based) protocols are a robust and scalable communication paradigm to disseminate information in a large-scale distributed environment using randomised communication [1]. The advantages of epidemic protocols over global

Pasu Poonpakdee · Giuseppe Di Fatta
School of Systems Engineering, University of Reading, Whiteknights,
Reading, Berkshire, RG6 6AY, United Kingdom
e-mail: p.poonpakdee@pgr.reading.ac.uk, g.difatta@reading.ac.uk

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_31

communication schemes based on deterministic interconnection overlay networks are their inherent robustness and scalability.

Applications based on epidemic protocols are emerging in many fields, including Peer-to-Peer (P2P) overlay networks (e.g., [2]), mobile ad hoc networks (MANET) (e.g., [3]) and wireless sensor networks (WSN) (e.g, [4]). More recently, epidemic protocols have been adopted to support fully decentralised data mining tasks for extreme-scale systems [5, 6], even under node churn and network failures [7].

In general, Epidemic protocols can be adopted for information dissemination and to solve the distributed data aggregation problem in a fully decentralized manner. The goal of data aggregation in networks is the parallel determination at each node of the exact value, or of a good approximation, of a global aggregation function over a distributed set of values. In this work, global aggregation is used as a typical application for the performance analysis of epidemic protocols.

The expansion property of graphs is a fundamental mathematical concept [8], which has been adopted to study several aspects of complex networks and social graphs [9, 10], and quasirandom rumor spreading was shown to exhibit a natural expansion property [11].

This work investigates the expansion property of the overlay topologies induced by epidemic protocols. The expansion quality of epidemic membership protocols is introduced and simulations show that it is a good indicator of the convergence speed of an epidemic global aggregation.

The rest of the paper is organised as follows. Section 2 introduces the adopted definition and measures of the expansion property of graphs. In Section 3 epidemic protocols are briefly reviewed. Section 4 introduces a novel expander membership protocol. Section 5 presents the results of simulations and their analysis. Finally, Section 6 provides some conclusive remarks and directions of future work.

2 Expansion Property of Graphs

Expander graphs, or simply expanders, are sparse graphs that have strong connectivity properties. Informally, a graph is an expander if any vertex subset (not too large) has a relatively large set of one-hop distant neighbours.

There are a few alternative definitions of expanders, which are based on the vertex expansion, the edge expansion or the spectral gap. The definition of expanders adopted in this work, is based on the vertex expansion.

Given a graph $G = (V, E)$, where V is the set of nodes and E the set of edges, and a subset of nodes $S \subset V$, the outer boundary of S is the set of nodes that are not in S and have at least one neighbour in S. Specifically, the outer boundary is defined as:

$$\partial(S) = \{v \in V \backslash S : \exists u \in S \mid \langle u, v \rangle \in E\}. \quad (1)$$

The *expansion ratio* of a subset $S \subset V$ is defined as $\frac{|\partial(S)|}{|S|}$. Although the typical measure of expansion quality of a graph is based on the *expansion ratio*, it is sometimes convenient to adopt some normalised variant [9]. The relative size of the outer boundary of a subset $S \subset V$ is here adopted to define the **vertex expansion index** $h(G, S)$ as:

$$h(G, S) = \frac{|\partial(S)|}{|V \backslash S|}. \quad (2)$$

The vertex expansion index is defined in $[0, 1]$ regardless of the graph order ($|V|$) and the sample size ($|S|$), while the *expansion ratio* is not.

For a given sample size s, the minimum and maximum vertex expansion indices are defined as:

$$h_{min}(G, s) = \min_{S \subset V, |S|=s} \frac{|\partial(S)|}{|V \backslash S|} \; and \quad (3)$$

$$h_{max}(G, s) = \max_{S \subset V, |S|=s} \frac{|\partial(S)|}{|V \backslash S|}. \quad (4)$$

These two expansion indices measure the range of the outer boundary cardinality for a given sample size with respect to the largest possible outer boundary.

Expanders are typically characterised by the minimum value of the expansion property over a specific range of the sample size, i.e. $0 < |S| \leq \frac{|V|}{2}$ However, in this work, we have adopted a fixed sample size to carry out the analysis. In a preliminary analysis, we have experimentally determined that a sample size of 5% of the order of the graph is a good choice, as larger samples may reach the largest possible expansion.

3 Epidemic Protocols

Epidemic protocols are typically described as periodic and synchronous with a cycle length Δ_T and are executed for a number of cycles T_{max}. At all discrete times t ($0 < t \leq T_{max}$), each node independently sends information to a peer, which is ideally selected uniformly at random among all nodes in the system. This selection operation is provided by a peer sampling service, the *membership protocol*, which is implemented with an epidemic approach.

Epidemic membership protocols are necessary to support application-level protocols, which provide services such as decentralised data aggregation. At each cycle, a membership protocol defines an overlay topology, over which communication operations of the application-level service are based.

Membership protocols and aggregation protocols are briefly reviewed in the following sections.

3.1 Membership Protocols

The node sampling service is considered a fundamental abstraction in distributed systems [12]. In large-scale systems, nodes cannot build and maintain a complete directory of memberships. A membership protocol builds and maintains a partial view of the system, which is used to provide the random node selection service. The distributed set of views implicitly defines an overlay topology $G = (V, E)$. A membership protocol periodically and randomly changes the local views, thus generating a sequence of random overlay topologies $\Gamma = \{G_i\}$, with $G_i = (V_i, E_i)$ being the overlay topology at protocol cycle i.

The required assumptions are that the physical network topology is a connected graph, a routing protocol is available and an initialisation mechanism for the overlay topology is provided.

Several membership protocols have been proposed in the literature, which have typically been designed to produce random overlay topologies. However, none of the previous approaches has considered the expansion quality of the induced overlay topologies as design principle, nor to analyse the performance.

The *Node Cache Protocol* [13] is the simplest membership protocol, which adopts a straightforward approach based on a symmetric *push-pull* mechanism. At each node, the protocol maintains a local cache Q of node identifiers (IDs), with $|Q| = q_{MAX}$. At each protocol cycle, the content of the local cache is sent (*push*) to a node randomly chosen from the local cache according to a uniform probability. When a remote cache is received, the local cache is sent (*pull*) to the remote node. The remote cache and the remote node ID (refresh mechanism) are merged with the local cache, which is finally trimmed to the maximum size by randomly removing the number of IDs exceeding q_{MAX}. The node cache protocol provides a local service which approximates a random peer sampling in the global system. When invoked, the service removes and returns a random node from the local cache.

The protocol *Send&Forget* [14] is based on a simple *push* mechanism. At each protocol cycle, a portion of the local cache is sent to a node randomly chosen from the local cache according to a uniform probability. When a remote cache is received, it is merged with the local cache.

Cyclon [15] is a membership protocol that is an enhancement of a basic shuffling mechanism similar to the one adopted in the Node Cache Protocol. Cyclon adopts a lifetime (age) of the node IDs in the local cache and the selection of entries in the local cache is biased by their lifetime.

The protocol *Eddy* [16] attempts to minimize temporal and spatial dependencies between nodes' caches in order to provide a better random distribution of the node samples in the overall system. Eddy is arguably the most complex membership protocol and may incur in significant communication overhead.

3.2 Aggregation Protocols

The data aggregation problem refers to the computation of a global aggregation function in a network of nodes, where each node is holding a local value. Examples of global aggregation functions are the sum, the average, the maximum, the minimum, random samples, quantiles, etc. Local approximations of the global aggregate function can be obtained with an epidemic aggregation protocol.

Nodes periodically exchange their local state and the reception of a remote state triggers the update of the local state at a node. The update produces a reduction of the variance in the estimates in the system until convergence. The definitions of local state, type of messages and update operation depend on the particular aggregation protocol and the target global function. A good approximation of the global aggregate function can be obtained at every node within a number of protocol cycles.

Epidemic aggregation protocols may typically employ *push*, *pull* or *push-pull* schemes. The Symmetric Push-Sum Protocol (SPSP) [13] combines the accuracy and simplicity of a push-based approach and the efficiency of the push-pull scheme. SPSP does not require synchronous communication with atomic operations; it achieves a convergence speed similar to the push-pull scheme, while keeping the accuracy of the push scheme. SPSP and the global average as target function have been used in the simulations presented in this work.

4 An Expander Membership Protocol

The membership protocol introduced in this section is the first attempt to directly exploit the concept of expansion in graphs.

Memberships protocols can be seen as a distributed implementation of multiple random walks and are used to generate random overlay topologies that are sparse and have strong connectivity. After sufficiently many protocol cycles the set of neighbours (or outgoing edges) of each node are expected to be uniformly distributed. A quick convergence to a random overlay topology is an important quality of membership protocols, as it affects the convergence speed of the applications.

These considerations have inspired a new membership protocol based on the concept of vertex expansion, as briefly outlined here. The protocol is based on the symmetric (*push-pull*) cache shuffling approach as described in section 3.1. At each cycle and at each node i, a destination node x_0 is randomly selected from the local cache Q_i. A *push* message m_i containing the local cache Q_i is sent to x_0. At the reception of the message, node x_0 computes the intersection of the local cache Q_{x_0} and the remote cache Q_i. The message is accepted if $|Q_{x_0} \cap Q_i| \leq T_{max}$, where T_{max} is a neighbourhood similarity threshold ($T_{max} \geq 0$). If the incoming *push* message is accepted, a *pull* message is sent to node i and the two caches are merged. Otherwise, $|Q_{x_0} \cap Q_i| > T_{max}$ and the message m_i is forwarded to a node x_1 randomly selected from the local cache of node x_0. This procedure is repeated up to a maximum number of hops (H_{max}).

This simple protocol aims at maximising the expansion quality of the overlay topology by swapping and merging cache entries between nodes with low neighbourhood similarity. Therefore, clusters of nodes with high neighbourhood similarity are expected to break up sooner than in the 1-hop push-pull scheme.

This protocol may require additional components to optimise other aspects of the membership management task. However, for the purpose of this work this membership protocol is suitable to show the relation between expansion quality and the convergence speed of a global aggregation task.

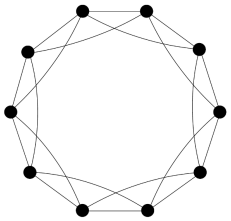


Fig. 1 Initial overlay topology: a regular circular lattice

5 Experimental Analysis

The goal of the experimental analysis is to compute measures of the expansion quality of the overlay topologies induced by different membership protocols and to investigate their relation with the convergence speed of an epidemic aggregation task in the network.

A membership protocol implements a distributed and continuous edge rewiring procedure to generate random overlay graphs. Edge rewiring is performed by means of random communication operations over the current overlay topology.

When a membership protocol is executed over a random regular graph, it generates a sequence of random regular graphs with similar characteristics. In this case, all membership protocols seem to provide an acceptable peer sampling service with respect to the convergence speed of the global aggregation. However, when the overlay topology is not a regular random graph, differences between protocols emerge. This may happen, for example, when the overlay topology is initialised (cold start) or when high churn introduces perturbations. Rather than studying optimal initialisation procedures, here we investigate the ability of a protocol to change a poor topology into a random graph quickly. This property is intuitively connected with the expansion quality of graphs, hence the concept of the expansion quality of a membership protocol.

A poor topology could be generated by a naive centralised initialisation procedure, e.g. a star topology, a small clique of servers to which all nodes are connected, or a circular lattice. They all have a poor expansion quality, and without the random rewiring mechanism of a membership protocol they would lead to a poor

performance of global aggregation and information dissemination protocols. Obviously, centralised topologies also suffer from load imbalance. In the experimental analysis, the overlay topology is initialised as circular lattice (Figure 1) with a constant out degree (30), which provides a poor expansion with a good load balance.

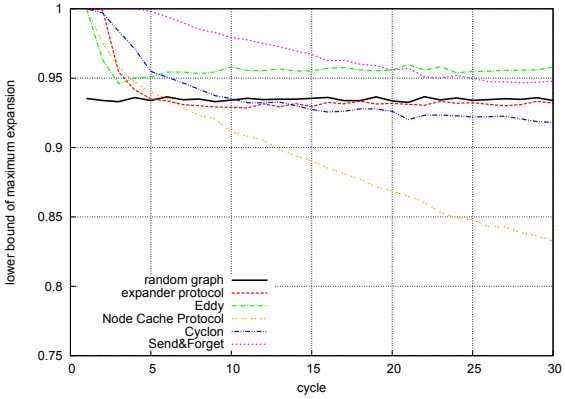
An asynchronous network configuration with a uniform distribution of network latencies has been adopted. The simulations have been executed with the following membership protocols, whose parameters have been set for best performance according to the literature and to a preliminary analysis:

- *Node Cache Protocol* [13] ($q_{max} = 30$),
- *Send&Forget* [14] with cache size upper and lower bounds of, respectively, 40 and 15,
- *Cyclon* [15] with shuffle length of 15,
- *Eddy* [16] with refresh rate of 10 cycles and shuffle length of 15,
- the novel expander protocol ($q_{max} = 30$, $H_{max} = 5$ and $T_{max} = 0$) and
- an ideal protocol based on random graphs with a constant outdegree of 30.

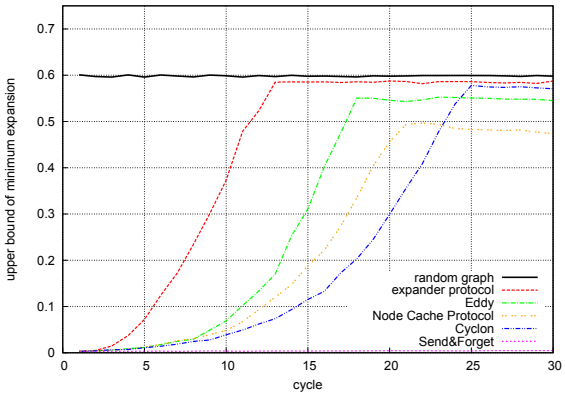
In all protocols the initial cache size (outdegree) is set to 30. The random membership protocol is included to provide a baseline performance and is based on the ideal global knowledge of the system to generate a different random graph at each cycle of the protocol.

In the first set of simulations (Figure 2), each protocol has been run for a number of cycles starting from the initial circular lattice topology. The expansion indices h_{min} and h_{max} for a sample size of 5% are used to monitor the evolution of the expansion quality of the overlay topology over the cycles. The exact values of the indices cannot be determined, as they would require an exhaustive search over a combinatorial number of node subsets. The approximation of the minimum and maximum expansion indices can be determined by a greedy algorithm. The adopted greedy algorithm is a variant of the one adopted in [9], where at step 6 the objective function $|N(S \cup \{v\})|$ is used in place of $|N(\{v\}) - (N(S) \cup S)|$. The different function can provide a tighter bound of the extreme values (min/max) when $v \in N(S)$. Although the values determined by the greedy method are just upper and lower bounds of the true extreme values, they are believed to be a good approximation.

In the charts of Figure 2, the minimum and maximum values of the vertical axis have been chosen to correspond to the average expansion index ($\bar{h} \approx 0.77$), which was determined by a Montecarlo method. Figure 2 shows that the maximum expansion index of the expander protocol converges quickly to the one of the random protocol. *Eddy* has a higher maximum and a lower minimum than the random protocol. The minimum expansion index of the expander protocol shows the fastest rate of convergence to the one of the random protocol, followed by *Eddy*, *Node Cache Protocol* and *Cyclon*. The minimum expansion index of *Send&Forget* remains close to 0 for the entire range of protocol cycles: in this case the overlay topology is not changing fast enough from the initial ring lattice. This test shows that the protocol that is explicitly designed to optimise the expansion quality of the overlay topology, as expected, achieves this goal better than the other membership protocols.



(a) maximum expansion index h_{max}



(b) minimum expansion index h_{min}

Fig. 2 Greedy approximation of the expansion indices (network size: 10000 nodes, sample size: 5%)

In the second set of simulations (Figure 3), the aggregation protocol *SPSP* [13] has been adopted to perform a global aggregation. The local values of the aggregation protocol are initialised with a peak distribution: all nodes have initial value 0, but one that has a peak value. After some protocol cycles the local values are expected to converge to the expected target value (global average). The standard deviation of the local aggregation values is used to measure the convergence speed of the aggregation protocol with the different membership protocols.

Figure 3 shows how the membership protocol can be relevant in terms of the convergence speed of a global aggregation task. The random protocol has a constant slope, which corresponds to a constant rate of the variance reduction in the system. The other membership protocols need to change the initial topology into a random graph before they can also provide a similar variance reduction rate (when the curves become parallel to the one of the random protocol). It is evident that the protocol explicitly based on the concept of expansion has the best performance. The protocol

Send&Forget has a particularly poor performance: it takes a very long time to rewire the topology into a random graph because it has an asymmetric communication pattern (*push* only) and rewires only a small number of edges for each message (shuffle length is 2).

The maximum expansion index (Figure 2(a)) does not seem to be a good indicator of the quality of the membership protocol. While the minimum expansion index (Figure 2(b)) is rather interesting: it clearly provides the same ranking of protocols as in Figure 3.

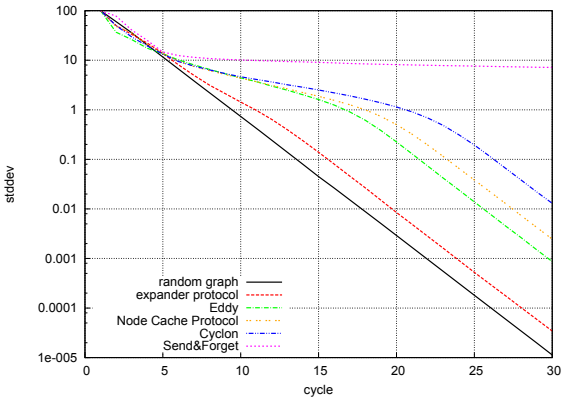


Fig. 3 Convergence speed of epidemic aggregation (network size: 10000 nodes)

Finally, the effect of different values for the parameters H_{max} and T_{max} of the expander protocol has been analysed, although the detailed results are not reported for the sake of brevity. T_{max} has a significant effect in the performance, with values closer to zero being the most effective. The effect of H_{max} is also important, though for $H_{max} > 2$ the improvement is less evident.

This work has shown that the minimum expansion index is useful to characterise epidemic membership protocols. Although evidence from more extensive simulations and experimental analysis in real-world networks would be required for more conclusive results.

6 Conclusions

This work has investigated how the convergence speed of decentralised epidemic aggregation may be affected by the underlying membership protocol. Membership protocols have the task of generating and evolving random overlay topologies to support fast epidemic aggregation and information dissemination. When the overlay topology is far from random, the membership protocol is expected to rewire the edges in such a way to quickly converge to random graphs. It has been shown that

different membership protocols perform this task at quite different rates, and, most importantly that the minimum expansion quality is a good candidate as indicator of this difference. Future work will be devoted to extend the experimental analysis to other initial topologies and network conditions, and in particular to study the effect of node churn.

References

1. Demers, A., Greene, D., Hauser, C., Irish, W., Larson, J., Shenker, S., Sturgis, H., Swinehart, D., Terry, D.: Epidemic algorithms for replicated database maintenance. In: Proc. of the Sixth Annual ACM Symposium on Principles of Distributed Computing, PODC 1987, pp. 1–12. ACM (1987)
2. Ghit, B., Pop, F., Cristea, V.: Epidemic-style global load monitoring in large-scale overlay networks. In: International Conference on P2P, Parallel, Grid, Cloud, and Internet Computing, pp. 393–398 (2010)
3. Ma, Y., Jamalipour, A.: An epidemic P2P content search mechanism for intermittently connected mobile ad hoc networks. In: IEEE GLOBECOM, pp. 1–6 (2009)
4. Chitnis, L., Dobra, A., Ranka, S.: Aggregation methods for large-scale sensor networks. ACM Transactions on Sensor Networks (TOSN) 4, 1–36 (2008)
5. Di Fatta, G., Blasa, F., Cafiero, S., Fortino, G.: Epidemic k-means clustering. In: Proc. of the IEEE Int.l Conf. on Data Mining Workshops, pp. 151–158 (2011)
6. Mashayekhi, H., Habibi, J., Voulgaris, S., van Steen, M.: GoSCAN: Decentralized scalable data clustering. Computing 95(9), 759–784 (2013)
7. Di Fatta, G., Blasa, F., Cafiero, S., Fortino, G.: Fault tolerant decentralised k-means clustering for asynchronous large-scale networks. Journal of Parallel and Distributed Computing 73(3), 317–329 (2013)
8. Hoory, S., Linial, N., Wigderson, A.: Expander graphs and their applications. Bullettin of the American Mathematical Society 43(4), 439–561 (2006)
9. Maiya, A.S., Berger-Wolf, T.Y.: Expansion and search in networks. In: Proc. 19th ACM Intl. Conference on Information and Knowledge Management, CIKM 2010 (October 2010)
10. Malliaros, F.D., Megalooikonomou, V.: Expansion properties of large social graphs. In: DASFAA International Workshop on Social Networks and Social Media Mining on the Web (SNSMW) (April 2011)
11. Doerr, B., Friedrich, T., Sauerwald, T.: Quasirandom rumor spreading: Expanders, push vs. pull, and robustness. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) ICALP 2009, Part I. LNCS, vol. 5555, pp. 366–377. Springer, Heidelberg (2009)
12. Jelasity, M., Voulgaris, S., Guerraoui, R., Kermarrec, A.M., van Steen, M.: Gossip-based peer sampling. ACM Trans. Comput. Syst. 25(3) (2007)
13. Blasa, F., Cafiero, S., Fortino, G., Di Fatta, G.: Symmetric push-sum protocol for decentralised aggregation. In: Proc. of the Int.l Conf. on Advances in P2P Systems, pp. 27–32 (2011)
14. Gurevich, M., Keidar, I.: Correctness of gossip-based membership under message loss. In: Proceedings of the 28th ACM Symposium on Principles of Distributed Computing, PODC 2009, pp. 151–160. ACM, New York (2009)
15. Voulgaris, S., Gavidia, D., Steen, M.: Cyclon: Inexpensive membership management for unstructured p2p overlays. Journal of Network and Systems Management 13(2), 197–217 (2005)
16. Ogston, E., Jarvis, S.A.: Peer-to-peer aggregation techniques dissected. Int. J. Parallel Emerg. Distrib. Syst. 25(1), 51–71 (2010)

Ontology and Rules-Based Model to Reason on Useful Contextual Information for Providing Appropriate Services in U-Healthcare Systems

Amina HameurLaine, Kenza Abdelaziz,
Philippe Roose, and Mohamed-Khireddine Kholladi

Abstract. In our days, the development of computing technologies has facilitated the daily life since the environment is improved by its different applications. Pervasive computing is one of the most efficient computing areas which offer several smart systems aiming to provide various significant services in our life. Context awareness, in general and context-aware adaptation in particular, is a central aspect of pervasive computing systems, characterizing their ability to adapt and perform tasks based on context. In this paper, we describe our proposal ontology and rules-based model for representing and reasoning on useful contextual information. This model permits to provide appropriate services in ubiquitous healthcare systems which are one of the main application areas for pervasive computing that allow monitoring the health and wellbeing of patients anytime and anywhere.

Keywords: Pervasive computing, U-Healthcare, context-awareness, context modeling, context reasoning, ontology, inferences rules.

1 Introduction

Ubiquitous or pervasive healthcare (U-Healthcare) systems are one of the main application areas for pervasive computing [1] that allow monitoring the health and

Amina HameurLaine
MISC Laboratory, Constantine 2 University, 25000 Constantine, Algeria

Kenza Abdelaziz
University Pierre and Marie Currie, Paris, France

Philippe Roose
LIUPPA / UPPA, 2 Allée du Parc Montaury 64600 Anglet, France

Mohamed-Khireddine Kholladi
El Oued University , MISC Laboratory of Constantine 2 University
e-mail: amina.hamerelain@hotmail.fr, abdelaziz.kenza@gmail.com,
Philippe.Roose@iutbayonne.univ-pau.fr, kholadi@yahoo.fr

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_32

wellbeing of patients anytime and anywhere; using intelligent environments technology which include sensors, services and smart mobile devices. This kind of system must provide us not only medical services but also services that allow controlling the daily activities of patient at his own home such as his shower, and controlling his physical environment such as home temperature. However, obstacles related to the limited resources of mobile devices like battery lifetime; privent the service continuity in this kind of systems which use mobile devices.

In order to provide the appropriate service to the user, the pervasive healtchare system must be able to gather contextual information over time which comes from different and heterogeneous sources, then to react rapidly by analyzing and reasoning dynamically not only on the health measurements of patient, but also on his current location, his profile, his physical environment and his devices'constraints. This makes context-awareness in general and context-aware adaptation in particular, an essential requirement for pervasive computing systems. One of the greatest challenges of pervasive computing is to model context information. Context modeling permits to represent contextual information and provide a high level of conceptual abstraction which allows reasoning on this information in order to adapt the behavior of system. Therefore, there is an increasing need to construct a uniform context model due to the diversity of contextual information sources. According to [8], ontologies seem to be one of the most suitable solutions for modeling context; due to their high and formal expressiveness and the possibilities for applying ontology reasoning techniques. ontology [13] can be defined as a formal explicit definition of a shared conceptualization which permit to provide general expressive concept and support for syntactic and semantic interoperability. In addition, their amenability for building context-aware pervasive computing systems is demonstrated in several works [14] [15] [16] [17]. In this context, we aim to propose a scalable ontology for modeling and reasoning not only upon patient's health measurements which can be gathered from different bio-sensors such as blood sugar measurement, but also for reasoning on all useful contextual information in order to provide the appropriate healthcare services and also the appropriate smart home service that allows patients and elderly persons to live safely and independently in their own homes. In addition, our ontology takes into consideration the limited resources of mobile devices and supports solutions that can be proposed by the developer for assuring the continuity of services, such as the migration of services into other devices or into the cloud when mobile devices cannot support services because of their resource constraints like battery lifetime.

The rest of the paper is organized as follows. Section 2 discusses some existing works. Section 3 presents our ontology and rules based-model. Finally, Section 4 concludes the paper and describes future work.

2 Related Works

Several interesting works are proposed in the field of pervasive healthcare system, which have demonstrated the amenability of ontologies for building context-aware pervasive computing systems. Catarinucci et al [14] have proposed a context-aware infrastructure for ubiquitous and pervasive monitoring of heterogeneous healthcare-related scenarios. Their system is based on ontology representation, multi-agent paradigm and rule-based logic. They have used an ontology and inference rules for modeling and reasoning on context information. In the work [15], authors proposed a u-healthcare service system and an ontology-based healthcare context information model to implement a ubiquitous environment. In the other work [16] an ontological model for organizing the knowledge in the heterogeneous domain of embedded devices and complex healthcare systems has been proposed. This ontology covers the domain of medical services where the medical services cover many areas such as patient care, clinical and administrative decisions, assisting devices and patient diagnostics. In [17] authors have proposed an intelligent context-aware system based on ontology which can be reused in any system uses bio context in pervasive environment. However, these works have not taken into consideration the limited resources of mobile devices such as battery lifetime of smart phone. In addition, all these works are based on healthcare or medical services. They don't take into account other interesting services that allow patients and elderly persons to live safely and independently in their own homes such as controlling home temperature and water temperature.

3 Our Ontology and Rules-Based Context Model for Pervasive Healthcare Systems

In our work, we propose ontology and rules-based model for representing and reasoning upon useful contextual information that must be taken into consideration for providing appropriate services in order to allow monitoring the healthcare of patient anywhere and anytime. Our ontology takes into account the limited resources of mobile devices and supports solutions that can be proposed by the developer to ensure the service continuity, such as the migration of services into other devices or into the cloud; when mobile devices cannot support services because of their resource constraints as battery lifetime. Before describing our ontology, we take a simple motivating scenario.

3.1 Scenario

Mr. Adem is an elderly diabetic person who lives alone at home and needs to monitor his health. He uses an intelligent environment which includes smart devices, sensors and services that are deployed on his smart phone. This intelligent environment constructs a pervasive healthcare system that allows him to live safely and independently within his own home. Let's take some examples of services of this system:

When Mr Adem needs to check his blood sugar level, he uses a bio-sensor connected to his smart phone via Bluetooth technology. According to his blood sugar level, the system will provide him the appropriate service. If his blood sugar level is out of the normal range determined by his doctor, there will be three possible cases:

- *High Blood Sugar level:* in this case the health situation of person is not in danger, but he has a high blood sugar level and he must take an insulin injection. The system will execute the "AdjestingInsulinDose" service which adjust automatically the insulin dose according to the blood sugar measurement, and execute the "Diabetes-Guide" which contains a guide about what must do as exercises and diet; for keeping his blood sugar level in normal.

- *Low Blood Sugar level:* in this case, the system will execute the "Diabetes-Guide" which contains a guide of what he must to do for adjusting his blood sugar level.

- *Danger Blood Sugar:* in this case, his health situation is in danger, he must be transferred to the nearest emergency center or hospital. The "Emergency" service must be executed in order to contact the hospital and sends a brief report contains the health situation of person and his personal information.

Mr Adem wants to take a shower, when he inters in his bathroom and activates the faucet; "Adjusting water Temperature" service will be executed automatically for adjusting the temperature of water.

At 20 o'clock, he should take his dinner and his drugs, but he often forgets it. For that, the system executes the "Drug Reminder" service for reminding him about the list of drugs and foods that can be taken in dinner.

3.2 Context Modeling

Context information can be characterized as static or dynamic [7]. Static context information describes the invariant information that can be obtained directly from users such as personal information. Dynamic context information describes the variant information that can be captured from different sensors such as blood pressure and location. Ubiquitous healthcare systems focus generally on dynamic context since they need to adapt their behaviors dynamically according to the change of this context. However, that does not prevent to take into account the static context such

as chronic disease for reasoning on health situation of patient. In this context, we construct an ontology that can be used for modeling useful contextual information (static and dynamic context) in pervasive healthcare systems. In fact, Ontologies are used to capture knowledge about some domain of interest; like in our works; pervasive healthcare domaine. Our ontology describes the concepts in this domain and also the relationships that hold between those concepts.

Ontology Classes

In ontology, classes are a concrete representation of concepts. As shown in Fig. 1, our ontology is composed of four general classes that represent cocepts of useful contextual information that are necessary for monitoring the healthcare situation of elderly persons in their smart homes or anywhere and anytime. These classes are; "Personal Data" class, "Sensor Data" class, "Services" class and "Host" class.

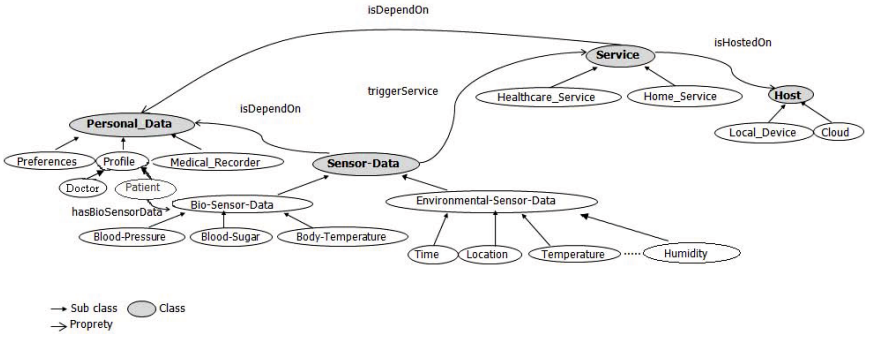


Fig. 1 Our Ontology structure

- The *Personal Data* class represents user's personal information that can be specified manually by the user; such as his profile, his medical information and his preferences.
- The *Sensor Data* class represents data that can be gathered from different sensors. We have two types of sensor data: i) Bio-Sensor Data represents data captured by Bio-Sensors, like blood pressure, blood sugar and body temperature. ii) Environmental Sensor Data represents data captured by environmental sensors, like time, location, temperature, etc.
- The *Services* class represents services that can be provided by the pervasive healthcare system. As we discussed in previous sections, pervasive healthcare systems must provide not only healthcare services, but also provide services that allow patients to live safely and independently in their own homes. For that we have created two sub classes of Service class; Healthcare Service class represents

Table 1 Exemples of Object Properties

Object Property	Source class (Domain)	Target class (Range)
triggerService	Sensor-Data	Service
isDependOn	Sensor-Data Service	Personal-Data
isHostedOn	Service	Host
useHost	Patient	Host
hasBioSensorData hasBloodSugar hasBloodPressure	Patient	Bio-Sensor-Data
hasMedicalRecorder hasChronicDisease hasDrugs	Patient	Medical-Recorder

all healthcare services such as Diabetes Services, Drug Service, Emergency Service, etc, and Home Service class represents services that allow monitoring the daily life of patients in thier smart home such as AdjustingWaterTemperature Service. All these services can be triggered by Sensor Data.

- The *Host* class represents different hosts of services proposed by developers. For instance, services can be hosted on Local Devices or on Cloud. The Local Device class contains all information about Fixed Devices and Mobile Devices. Mobile Devices have limited resources such as battery, memory and CPU. The Cloud class contains information about the cloud that can be used for hosting services.

Ontology Relationshipes

Relationships indicate the interaction among the concepts in the ontology. They are defined by the properties and by the attributes that characterize the classes. Relationships that hold between classes are called "Object properties". Each object property has source and target. Table 1 represents examples of object properties and their source/target classes that we had defined in our ontology. Some of these properties have sub-properties, e.g "hasBloodSugar" property is a sub-property of "hasBioSensorData". Other object properties such as "isMegratedOn" property can be inferred by using inference rules. Attributes that characterize the classes or the instances of classes are called "data type properties". They describe relationships between instances (individuals) of classes and data values. Table 2 presents examples of data type propreties. Like object properties, these properties can also have sub-properties such as "hasBloodSugarValue" which is sub-property of "hasBiosensorDataValue" data type property.

Ontology Language

The literature offers many languages to represent or express ontologies, including resource description framework schema (RDFS), DAML+OIL, and OWL. OWL is a key to the semantic web and was proposed by the Web Ontology Working Group of W3C [18]. OWL is a general purpose ontology language that contains all the necessary constructors to formally describe most of the information management definitions: classes and properties, with hierarchies, range and domain restrictions; that's whay we have used OWL language for describing our ontology.

Example of OWL Language

```
</owl:Ontology>
    </owl:Class>
    <owl:Class rdf:ID="Bio_Sensor_Data">
        <rdfs:subClassOf rdf:resource="#Sensor_Data"/>
    </owl:Class>
    <owl:Class rdf:ID="Blood_Pressure">
        <rdfs:subClassOf rdf:resource="#Bio_Sensor_Data"/>
    </owl:Class>
    <owl:Class rdf:ID="Blood_Sugar">
        <rdfs:subClassOf rdf:resource="#Bio_Sensor_Data"/>
        ..............
```

Table 2 Exemples of Data type Properties

Class	Data type Property	Data Type
Profile	hasProfileInformation hasName hasDateOfBirth	 string date .....
Bio-Sensor-Data	hasBiosensorDataValue hasBloodSugarValue	integer
Bio-Sensor-Data	hasBiosensorDataLevel hasBloodSugarLevel	string
Blood-Sugar	isBeforMeal	Boolean
Mobile-Device	hasBatteryLevel	integer

Table 3 Exemples of SWRL Rules

Physician Rules	Blood-Sugar(?x)⋀Patient(?p)⋀ hasBloodSugar(?p,?x) ⋀hasChronicDisease(?p,Diabet-Type-1)⋀ hasBloodSugarValue(?x,?y) ⋀isBeforMeal(?x,true) ⋀ swrlb:greaterThanOrEqual(?y,500) ⟶ hasBloodSugarLevel(?x,"Danger")
Developer Rules	Blood-Sugar(?x)⋀Patient(?p)⋀ ⋀hasBloodSugar(?p,?x)⋀hasBloodSugarLevel(?x,"Danger") ⋀Emergency-Service(?s) ⟶ triggerService(?x,?s)
	Patient(?p)∧Mobile-Device(?mobile)∧useHost(?p,?mobile) ∧Service(?s)∧isHostedOn(?s,?mobile)∧ hasbBatteryLevel(?mobile,?batterylevel)∧ swrlb:lessThanOrEqual(?batterylevel,15) ∧Cloud(?cloud)∧useHost(?p,?cloud) ⟶ isMigratedOn(?s,?cloud)

3.3 Context Reasoning

The main objective of proposing our ontology is not only to represent useful contextual information, but also to allow reasoning on this information using inference rules of the form "if...then...". Inference rules are a set of rules which define a general mechanism for discovering and generating automatically new relationships between concepts, based on existing ones [19]. In our ontology we use a set of inference rules for deducing services that will be triggered by sensor data and then which service will be provided to the user. We have classified our rules in two categories; Physician Rules and Developer Rules.

- *Physician Rules* are a set of rules related to the health situation and they are specified by a doctor and not by a developer. These rules permit to reason on Bio-sensor Data for deducing the health situation of person.

Example 1; *IF* (Blood sugar value is more than 500 m/l) *THEN* (blood sugar level is Danger).

- *Developer Rules* are specified by the developer. The developer defines a set of rules that permit to adapt the behavior of system in order to provide appropriate services according to the situation of person and to assure the continuity of services.

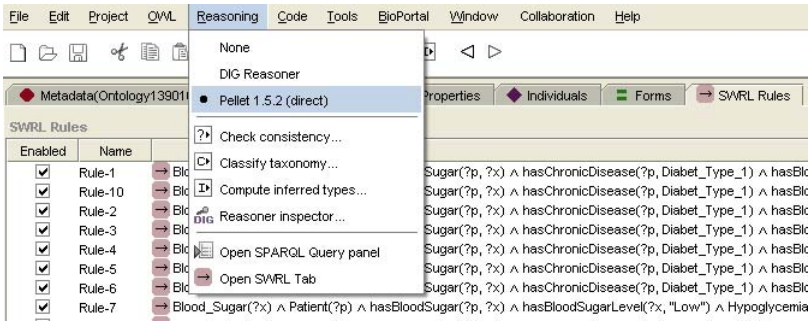


Fig. 2 Part of Our SWRL Rules on Protege

Example 2; *IF* (Patient has Danger blood sugar level) *THEN* (trigger Emergency service).

Example 3; *IF* (Service isHostedOn Mobile Device) *AND* (Battery Level is less than 15 percent) *THEN* (Service isMigratedOn Cloud).

For applying these rules on our ontology, we need to express them by a semantic language. The Semantic Web Rule Language (SWRL) is a proposed language for the Semantic Web that can be used to express rules as well as logic, combining OWL

with a subset of the Rule Markup Language. SWRL [20] extends the set of OWL axioms in order to include conditional rules (Horn clauses), of the form if... then.... In fact, a rule axiom consists of an antecedent and a consequent. In the "human-readable" syntax of SWRL, a rule has the form:

$$\text{antecedent} \Rightarrow \text{consequent}.$$

Informally, a rule may be read as meaning that if the antecedent holds (is "true"), then the consequent must also hold. Table 3 represents some examples of our SWRL rules used in our work.

As represented previously, we can perceive that our ontology is not complex to implement and can cover all useful cotextual information. In addition, our ontology and rules-model is scalable because we can easily add new contextual information and reasoning upon this new information, for instance, if a new sensor is added to the environment, like a heartbeat sensor, we can add this information by just changing the ontology structure and creating new rules for this information.

3.4 Ontology Implementation

For building our ontology and SWRL rules, we have used the Protégé tool [21]. Protégé is a free, open-source platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies. At its core, Protégé implements a rich set of knowledge-modeling structures and actions that support the creation, visualization, and manipulation of ontologies in various representation formats including OWL, RDF(S), and XML Schema. Using SWRL Tab and Pellet reasoner [22], we have created and tested our rules on the ontology; as showen in Fig.2.

4 Conclusion

Ubiquitous healthcare systems are one of the main application areas for pervasive computing which aim to provid several services; that allow monitoring the health and wellbeing of patients anytime and anywhere. For being able to provide us such service, this kind of systems need to adapt themselves automatically in response to the dynamic change of context. One of the greatest challenges of pervasive computing is to model context information due to the diversity of context information sources. Conciquantly, ther is an increasing need to construct a uniform context model that allows representing and reasoning upon useful contextual information. This paper presents a scalable ontology and rules-based model for representing and reasoning not only upon patient's health measurements which can be gathered from different Bio-Sensors, but also for reasoning on all useful contextual information that

must be taken into consideration for providing the appropriate healthcare and smart home services. In future work, we plan to integrate our ontology in a real context-aware system that provides different services in pervasive healthcare environment in order to evaluate this ontology with different and heterogeneous entities such as bio-sensors and environmental sensors.

References

1. Catarinucci, L., Colella, R., Esposito, A., Tarricone, L., Zappatore, M.: Rfid sensor-tags feeding a context-aware rule-based healthcare monitoring system. Journal of Medical Systems, 3435–3449 (2012)
2. Fensel, D.: Ontologies: A silver bullet for knowledge management and electronic commerce, Berlin (2001)
3. Henricksen, K., Indulska, J., Rakotonirainy, A.: Modeling context information in pervasive computing systems. In: Mattern, F., Naghshineh, M. (eds.) PERVASIVE 2002. LNCS, vol. 2414, pp. 167–180. Springer, Heidelberg (2002)
4. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.: Swrl: A semantic web rule language combining owl and ruleml. W3C Member Submission (2004)
5. Kim, J., Chung, K.: Ontology-based healthcare context information model to implement ubiquitous environment. Springer US (2011)
6. Ko, E.J., Lee, H.J., Lee, J.W.: Ontology-based context-aware service engine for u-healthcare. In: The 8th International Conference on Advanced Communication Technology, ICACT 2006, vol. 1, pp. 632–637 (2006)
7. Smith, M., Welty, C., McGuinness, D.: Owl web ontology language guide, `http://www.w3.org/TR/owl-guide/` (last access: February 2014)
8. Strang, T., Linnhoff-Popien, C.: A context modeling survey. In: The Sixth International Conference on Ubiquitous Computing, England (2004)
9. Thomas, J.: Protégé, `http://protegewiki.stanford.edu/wiki/Main_Page` (last access: February 2014)
10. W3C: Inference, `http://www.w3.org/standards/semanticweb/inference` (last access: February 2014)
11. W3C: Pellet, `http://www.w3.org/2001/sw/wiki/Pellet` (last access: February 2014)
12. Weiser, M.: The computer of the twenty-first century. Scientific American, 94–100 (1991)
13. Zeshan, F., Mohamad, R.: Medical ontology in the dynamic healthcare environment. Elsevier Ltd. (2012)

Following the Problem Organisation: A Design Strategy for Engineering Emergence

Victor Noël and Franco Zambonelli

Abstract. To support the development of self-organising systems, we explain and rationalise the following architectural strategy: directly mapping the solution decomposition on the problem organisation and only relying on the problem abstractions for the design. We illustrate this with an example from swarm robotics.

1 Introduction

Complex systems are made of simple elements and are characterised by the presence of non-linear interactions between them, no central control and the appearance of emergent behaviours at the system level [12]. In particular, emergence is the appearance of high-level behaviours resulting from low-level simpler rules [5] and an important mechanism governing these systems is self-organisation: autonomous change of the elements organisation without external control [5]. Multi-Agent Systems (MAS) is one field where self-organisation and emergence are studied and applied to engineer self-adaptive systems [6]. Some aspects of the global functionality are not explicitly pre-designed but emerge at runtime through this self-organising process in an endogenous and bottom-up way: the agents are unaware of the organisation as a whole [14]. Here, we assume self-organisation as the principle followed to design the low-level rules that lead to emergence.

In this paper, the general challenge is engineering self-adaptive self-organising complex systems that exist in and modify a complex context while meeting complex requirements, in the continuation of [1, 4].Towards that goal, we propose to look at practical methodological guidelines to accompany their design. In the following, we use the term "Self-Organising MAS" (SOMAS) to denote such engineered system.

Victor Noël · Franco Zambonelli
DISMI, University of Modena and Reggio Emilia, Italy
e-mail: victor.noel,franco.zambonelli@unimore.it

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_33

The contribution of this paper is proposing and rationalising the following design strategy: when designing SOMAS, it is necessary to follow the problem organisation. It means mapping the SOMAS decomposition on the problem decomposition in elements (and not sub-problems), and relying only on the problem abstractions for the behaviours. This strategy is not a method by itself but a complement to existing approaches and methods. It is illustrated with a running example: a swarm of bots exploring and rescuing victims in an unknown environment. Everything presented is fully implemented (See `http://www.irit.fr/~Victor.Noel/unimore-ascens-idc-2014/`).

2 Following the Problem Organisation

2.1 Problems, Requirements and Design Constraints

A problem to answer is made of a context and requirements: engineering is finding a software solution, here the SOMAS, satisfying the requirements in that context [10].

As an example, in a robotics scenario, we look at the search and secure problem: bots must explore an unknown environment to look for victims and then secure them (supposedly to rescue them, but this is not covered in our example). The context is composed of the bots (controlled by the software to build) with limited communication capabilities, the environment that have walls, the victims that must each be secured by several bots. The requirements are to search and secure victims, to secure them all, as fast as possible, to explore the accessible space fairly, to completely explore the space in a non-random way, etc.

It is important to highlight the distinction between the problem answered by the choice of using self-organisation and emergence, and the design constraints it implies: existing works characterising self-organisation and emergence do not usually explicit this distinction [5, 6, 11, 14].

The following can be part of the problem: to have self-adaptation, a distributed deployment context, large-scale system, non-existence of an efficient centralised solution or impossibility of expressing the global behaviour of the system [3]. Inversely, the following are mandatory design constraints when engineering SOMAS: the fact that the decision must be distributed and decentralised, that the global macro-level behaviour and organisation can't be predefined or that self-organisation must be a bottom-up process initiated locally by the elements of the system [11].

2.2 The Strategy

When designing SOMAS, two main activities are of importance: decomposing the solution in agents and giving them a self-organising behaviour. It is usual in software

engineering to first model the problem space before entering the solution space [10]. However, here, we closely map the solution decomposition in agents on the problem organisation, and only rely on the problem abstractions to design their behaviours.

By problem organisation, we mean the identification of the various elements participating in it and of the role they play with respect to the requirements. We call it the "organisation" to avoid confusion with the meaning usually associated to a decomposition of the problem in sub-problems.

In the robotic example, the elements participating in the problem are the bots, but also the victims and the environment. Then, in relation with the requirements, the elements play the following role in the problem: bots moves to directions they choose, bots communicate with other bots, bots perceive victim, bots perceive walls, victims are situated, victims need a specific number of bots to be secured, etc. Inversely, an example of a potential decomposition in sub-problems would describe the problem as being about exploring and discovering on one hand, and securing collectively on the other hand.

The problem organisation can be imposed by the context (that the engineer can't control or modify) or must be chosen by the engineer when building the system. How to do so is an open question that we don't answer here.

Based on this modelling of the problem, we map elements of the solution (software agents) to the elements of the problem, and we give them the same capabilities as in the problem domain and not more. Their behaviour should be designed locally with respect to the relations elements have in the problem. The decisions (including those of their self-organising behaviour) they take should only rely on the problem domain abstractions and no higher-level global abstractions should be introduced.

In the robotic example, bots must choose where is the best direction to go at every given moment. For that, they can use what they directly see (victims and explorable areas), and when they don't know what to choose, they should rely on information shared by other bots about the state of the world with respect to the problem: where they are needed for victims or exploration. Hence, bots that see victims or explorable areas advertise about it. This information can be propagated by the bots and they can use it to decide where to go next.

Of course the complexity of the context and of the requirements (e.g., high number of bots, unknown scattering of the victims or limited perception means) are likely to make all these choices difficult. Correctly choosing the best action to take is thus an important questions: we don't pretend to answer it in this paper, but, as said before, we argue that such decisions must rely on the problem domain abstractions. Still, we comment on this question in the next section.

2.3 Relation to the Design of Self-organisation

The strategy presented in the previous section can thus be used to design a SOMAS, but, as we highlighted it, is not enough by itself. In particular, a very important point is the problem of taking the correct local decision for the agents. Some approaches

to self-organisation propose local criteria to be followed by the agents in order to drive the self-organisation. For example, the AMAS theory [9] is such an approach. Its main design strategy is that agents must have a cooperative social attitude: the whole approach rests on the theory that if the agents of a system are cooperative with the system environment as well as internally, then the system will behave adequately with respect to the objectives of the agents and of their environment. By identifying local non-cooperative situations agents can face, the engineer designs the agents so that they prevent or correct such situations in order to put the system in a state as cooperative as possible. Usually, a measure called criticality that is shared amongst agents is used to reflect the importance of some state of the problem and to give an agent a way to decide between several choices.

In the robotic example, a bot often has the choice between several directions and do not see any victims. In order to take the most cooperative decision, he needs some information about the state of the system: bots can advertise for example about the direction they chose to go to and some measure (the criticality) of how much more bots are needed in this direction. When a bot propagates this information (because he chose the direction), it will update this criticality in order to reflect his and others participations in the self-organisation process: its choice means that this direction is a bit less critical now. Because bots assume they are all cooperative, they know that a direction chosen by another bot is presently the most important one to go to: choosing the most critical direction amongst all the neighbouring advertisements is enough for a bot to decide where to go next. Every decision taken will then influence how the bot computes this criticality, and inversely.

The way the self-organising process can be designed with this approach heavily relies on the fact that the agents does not contain any pre-defined behaviour in relation to the expected global behaviour, but only concepts manipulated in the definition of the problem itself, which serves to base the local decision on. For example, the criticality measure used in the AMAS approach reflects some aspect of the problem state in a comparable form: no extra high-level characterisation of what is or not a good global solution is used.

2.4 Rationale

The rationale behind the defended strategy, namely to follow the problem organisation when designing a SOMAS, relies on the design constraints highlighted in Section 2.1 and can be discussed in two cases: why follow the problem for decomposition and why use the problem organisation as we defined it here.

If the the engineer of a SOMAS introduces extra concepts foreign to the problem organisation, this means that when facing local decisions, the agents must translate their interpretation of the current state of the problem to the extra abstractions. Going far from the problem implies that we pre-set how situations are interpreted by the agents: it prevents them from interpreting correctly unforeseen situations because the concepts they manipulate can't capture them. In other words, the farther the design

is from the problem, the lesser adaptive the system will be, and the lesser adequate behaviours can emerge.

In the robotic example, if the bots are designed so that to explore, they move in the direction of a repulsion vector from other bots (a typical algorithm for bots dispersion), then the problem solved is not about exploring while securing anymore, it is about dispersing bots in an environment: for example, in a hallway, a stopped bot securing a victim will prevent other bots to go behind him. Inversely, if bots behave as explained before, when the collective would profit from dispersing, then bots disperse as a result of going in directions advertised by others where the less bots are going and when there is only one direction to go (e.g., a hallway), bots just go there because it is the only advertised direction.

Then, a problem decomposition in sub-problems calls for solving each sub-problem separately (if it is not the case, then the decomposition in sub-problems is useless for the design and this is out of the scope of the discussion here). This means that the sub-solutions must then be integrated together in the agents, and such integration is embedding the complexity of the problem.

In the robotic example, if the bots have a behaviour to explore and discover victims, and another to go help other bots secure their discovered victims, it becomes very difficult to handle at the agent level the choice between going in a direction or not: it could be needed to secure a victim, but there may be already other bots going there, so it must gather information about that, and then it could not be needed because other bots are going, but then maybe there is more to explore behind the victim, so it should go anyway, except in the case where there is still enough bots going there for the same reason as it is, and so on...

This puts back the complexity of solving the problem at the agent level instead of making it the result of the collective behaviour: it is the very reason why the paradigm shift proposed by self-organisation and emergence engineering was proposed in the first place. This matter has been discussed many times in the literature (the "complexity bottleneck" [11]): we don't bring new arguments for it.

3 Related Works and Discussion

The question of engineering emergence has been studied in various contexts, we discuss some of them and show how they are different from our contribution. On the whole, there is three ways of engineering emergences: ad-hoc, reusing or following a well-defined methodological approached.

First, ad-hoc means there is no explicit rationale behind decisions taken during the design: this is clearly out of scope of the current discussion as we are interested in ways to improve the engineering of SOMAS.

Then, reusing is usually done through the reuse of existing self-organising mechanisms that are well-known and understood. The main example of that is nature-inspired self-organisation [8]. These works rely on approaches or mechanisms dependent to a certain class of problems: they are easier to apply and to reuse when

possible, but in exchange it is needed to translate the concepts manipulated in the problem to the abstractions of the solution reused. It is on that point that it diverges from our work: this means that part of the original problem is lost during that translation, and we precisely advocate for relying extensively on the problem.

Finally, methodological approaches are the closest to our work in terms of motivation. We cited in Section 2.3 the AMAS approaches and similar strategies are for example well discussed in [7]. All these works mainly proposes way to design the self-organising behaviour of the agents but mostly don't discuss (or rationalise) the decomposition in agents of SOMAS: this is what we do here. Other works note that simulation can be used to accompany the engineering of emergence in order to iteratively change the design with respect to the observed results: it is called "co-development" [2] or "disciplined exploration" [13]. Our contribution is well coherent with these approaches, but of a different nature, and show that it is possible to exploit the problem organisation to reduce the development effort of SOMAS.

4 Conclusion

In this paper, we defend the idea that the specificities of self-organisation and emergence rationalise the need for decomposing the system by following the problem organisation. Of course, such an absolute assertion must be instantiated differently depending on the actual problem to solve: at the very least, this strategy and rationale improve the understanding of the relation between the problem and the solution decomposition. There exist many more other issues to explore on this subject, like how to well model the problem organisation, and other related subjects, like how the problem and the solution decomposition impacts the decentralised decision making.

Acknowledgements. This work is supported by the ASCENS project (EU FP7-FET, Contract No. 257414).

References

1. Abeywickrama, D.B., Bicocchi, N., Zambonelli, F.: SOTA: Towards a general model for self-adaptive systems. In: WETICE Conference, pp. 48–53. IEEE (2012)
2. Andrews, P., Stepney, S., Winfield, A.: Simulation as an experimental design process for emergent systems. In: EmergeNET4 Workshop: Engineering Emergence (2010)
3. Arcangeli, J.P., Noël, V., Migeon, F.: Software Architectures and Multiagent Systems. In: Oussalah, M. (ed.) Software Architectures, vol. 2, pp. 171–208. Wiley (2014)
4. Cabri, G., Puviani, M., Zambonelli, F.: Towards a taxonomy of adaptive agent-based collaboration patterns for autonomic service ensembles. In: International Conference on Collaboration Technologies and Systems, pp. 508–515. IEEE (2011)
5. De Wolf, T., Holvoet, T.: Emergence versus self-organisation: Different concepts but promising when combined. In: Brueckner, S.A., Di Marzo Serugendo, G., Karageorgos, A., Nagpal, R. (eds.) ESOA 2005. LNCS (LNAI), vol. 3464, pp. 1–15. Springer, Heidelberg (2005)

6. Di Marzo Serugendo, G., Gleizes, M.P., Karageorgos, A.: Self-organisation and emergence in mas: An overview. Informatica 30, 45–54 (2006)
7. Di Marzo Serugendo, G., Gleizes, M.P., Karageorgos, A. (eds.): Self-Organising Software. Natural Computing. Springer (2011)
8. Di Marzo Serugendo, G., Karageorgos, A., Rana, O.F., Zambonelli, F. (eds.): ESOA 2003. LNCS (LNAI), vol. 2977. Springer, Heidelberg (2004)
9. Georgé, J.P., Gleizes, M.P., Camps, V.: Cooperation. In: Di Marzo Serugendo, et al. (eds.) [7], pp. 193–226
10. Hall, J., Rapanotti, L., Jackson, M.: Problem-oriented software engineering: Solving the package router control problem. Transactions on Software Engineering 34(2), 226–241 (2008)
11. Heylighen, F., Gershenson, C.: The meaning of self-organization in computing. IEEE Intelligent Systems, Section Trends & Controversies 18(4), 72–75 (2003)
12. Mitchell, M.: Complexity: A guided tour. Oxford University Press (2009)
13. Paunovski, O., Eleftherakis, G., Cowling, T.: Disciplined exploration of emergence using multi-agent simulation framework. Computing and Informatics 28(3), 369–391 (2009)
14. Picard, G., Hübner, J.F., Boissier, O., Gleizes, M.P.: Reorganisation and Self-organisation in Multi-Agent Systems. In: International Workshop on Organizational Modeling, pp. 66–80 (2009)

Part X
Parallel Computing

Core Heuristics for Preference-Based Scheduling in Virtual Organizations of Utility Grids

Victor Toporkov, Anna Toporkova, Alexey Tselishchev,
Dmitry Yemelyanov, and Petr Potekhin

Abstract. Distributed environments with the decoupling of users from resource providers are generally termed as utility Grids. The paper focuses on the problems of efficient scheduling in virtual organizations (VOs) of utility Grids while ensuring the VO stakeholders preferences. An approach based on the combination of the cyclic scheduling scheme, backfilling and several heuristic procedures is proposed and studied. Comparative simulation results are introduced for different algorithms and heuristics depending on the resource domain composition and heterogeneity as well as on the VO pricing policy. Considered scheduling approaches provide different benefits depending on the VO scheduling objectives. The results justify the use of the proposed approaches in a broad range of the considered resource environment parameters.

1 Introduction

In distributed environments with non-dedicated resources such as utility Grids the computational nodes are usually partly utilized by local priority jobs coming from resource owners. Thus, the resources available for use are represented with a set of slots - time intervals during which the individual computational nodes are capable

Victor Toporkov · Dmitry Yemelyanov · Petr Potekhin
National Research University "MPEI", ul. Krasnokazarmennaya, 14, Moscow, 111250, Russia
e-mail: {ToporkovVV,YemelyanovDM,PotekhinPA}@mpei.ru

Anna Toporkova
National Research University Higher School of Economics, Moscow State Institute of Electronics and Mathematics, Bolshoy Trekhsvyatitelsky per., 1-3/12, Moscow, 109028, Russia
e-mail: AToporkova@hse.ru

Alexey Tselishchev
European Organization for Nuclear Research (CERN), Geneva, 23, 1211, Switzerland
e-mail: Alexey.Tselishchev@cern.ch

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_34

to execute parts of independent users' parallel jobs. The presence of a set of slots that generally have different start and finish time and difference in performance impedes the problem of coordinated selection of the resources necessary to execute the job flow from computational environment users. Resource fragmentation also results in decrease of the total computing environment utilization level. In such conditions, resource management and job scheduling based on economic models may be considered as an efficient way to take contradictory preferences of computing participants into account [1, 2].

Application-level scheduling (single user-based resource brokering [1]) with diverse optimization criteria set by independent users, as a rule, does not imply any global resource sharing or allocation policy. The regulations of a virtual organization (VO) in Grid usually imply a job flow scheduling. A metascheduler or a metabroker is considered as intermediate link between the users and the local resource management and job batch processing systems [1–4]. VOs naturally restrict the scalability of resource management systems (though, it is worth noting here, that there is a good experience [4] of enabling interoperability among metaschedulers belonging to different VOs). However, uniform rules of resource sharing and consumption established in VO make it possible to improve the job-flow scheduling efficiency. Metascheduling of different applications from a community of users in VOs of utility Grids aims to address this scheduling problem. In [5], a generalization of our original cyclic scheduling scheme (CSS) [6] with Batch-slicer (BS) is proposed. CSS implements job-flow scheduling in cycles by separate job batches on the basis of dynamically updated local schedules of computational nodes. BS implies "slicing" of an initial job batch into a set of sub-batches and each sub-batch scheduling.

A main contribution of this paper is as follows. First of all, we address a problem of early resources releases and rescheduling "on the fly" combining BS [6] and backfilling [7]. For the overall job-flow execution optimization and a resource occupation time prediction existing schedulers rely on the time specified in the job request, e.g. using Job Submission Description Language (JSDL). However, the reservation time is usually based on user inaccurate runtime estimates. In case when the application is completed before the term specified in the resource request, the allocated resources remain underutilized. Second, we introduce a heuristic of schedule shifting (Shifted CSS) in order to prevent the resource fragmentation.

Thus, we outline two main job-flow optimization directions. In the first of them, the optimal or suboptimal scheduling under a given criterion or criteria specified in VO, is performed on the basis of a priori information about local schedules of computational nodes and the resource reservation time for each job execution. CSS belongs to this type of systems. Another approach represents scheduling "on the fly" depending on a dynamically updated information about resource utilization. In this case, schedulers are focused on overall resources load maximization and job start time minimizing. Backfilling may be related to this type of scheduling. In [5], we introduce a combined approach. During every scheduling cycle a set of high priority jobs is grouped into a separate sub-batch. The scheduling of this sub-batch is further performed by BS. The scheduling of the rest of the batch jobs is performed by

backfilling with the dynamically updated information about the actual computational nodes utilization. The cyclic scheduling method combined with backfilling (Batch-slice-Filling - BSF) unites the main advantages of both BS and backfilling (BF).

The paper relates to comparative investigation of scheduling efficiency indicators using the initial CSS, Shifted CSS, BSF and BF depending on the number and heterogeneity of computational nodes as well as on the pricing policy of the VO.

The rest of the paper is organized as follows. Section 2 presents brief analysis of related works. In Section 3, there is a concept of a CSS combining backfilling and shifting. Section 4 contains simulation results of comparison of diverse scheduling approaches. Finally, section 5 summarizes the paper and describes further research topics.

2 Related Works

Many resource selection and scheduling algorithms, and heuristic-based solutions have been proposed for parallel jobs and tasks with dependencies in distributed environments [1,3,4,8–19]. In some well-known models of distributed computing with non-dedicated resources, only the first fit set of resources is chosen depending on the environment state [9–12], while job scheduling optimization mechanisms are usually not supported. In other models [3, 8, 19] the aspects related to the specifics of environments with non-dedicated resources, particularly dynamic resource loading, the competition between independent users, user global and owner local job flows, are not considered. In [8] heuristic algorithms for slot selection based on user-defined utility functions perform slot window allocation under the maximum total execution cost constraint, but the optimization occurs only on the stage of the best-found offer selection. Architecture and an algorithm for performing Grid resources co-allocation without the need for advance reservations based on synchronous queuing of subtasks is proposed in [14]. However, it is well-known, that advance reservation is effective to improve the co-allocation quality of service. Advance reservation-based co-allocation algorithms are proposed in [10–12, 15, 16].

First fit resource selection algorithms [9–12] assign any job to the first set of slots matching the resource request conditions without any optimization. Preference-based matchmaking [9] is not focused on the scheduling process. The job is scheduled on the first available resource according with user preferences. In [4], an approach to resource matchmaking among VOs combining hierarchical and peer-to-peer models of meta-schedulers is proposed.

The algorithms described in [15–17] imply an exhaustive search. Approaches in [16, 17] are based on a linear integer programming [16] or mixed-integer programming model [17]. In [16], users can specify a time frame for each resource: the earliest start time, the latest start time, and the job duration, where the user wants to reserve a time slot. This condition imposes restrictions for slots search only within this time frame. A linear integer programming-driven model with a genetic

algorithm is proposed in [1]. It allows obtaining the best meta-schedule that minimizes the combined cost of all independent users in a coordinated manner. In [17], the mixed-integer programming model is proposed. It performs the best scheduling in environments composed of multiple clusters that act collaboratively. The scheduling approaches in [1, 15–18] are efficient under given criteria: the processing cost, the overall makespan, resources utilization, load balancing for related tasks [18], etc. It is worth mentioning here, that complexity of the scheduling process is extremely increased by the resources heterogeneity and the task co-allocation process for parallel jobs across resource domain boundaries.

In this work, algorithms for efficient slot selection based on user, resource owner and VO administrator preferences with the linear complexity on the number of all available time-slots are used [13]. In our approach, users may introduce criteria into a job request format, e.g. JSDL, for slot search algorithms. BSF takes into account preferences of diverse VO stakeholders. Scheduling optimization is conducted on two levels - when selecting the slots and when executing the job batch.

3 A Concept of BSF Combined with Heuristics

The BSF (Fig. 1) procedure has two main steps. Firstly, the initial sub-batch of high priority jobs is scheduled by BS. Secondly, the scheduling of the subsequent sub-batch of lower priority jobs is performed by backfilling. Slot costs $C_j - 1, C_j, C_j + 1$ in Fig. 1 are determined by resource owners.

The preferences of the VO administrators are taken into account during the first BSF step. First of all, a set of non-intersecting alternatives (slot "windows") is allocated for each batch job on the scheduling interval. Second, the dynamic programming methods [13] are used to choose an optimal alternatives combination (with respect to the given criterion and with a restriction to a total batch job execution budget). This combination forms the final BS schedule. In order to satisfy the user preferences in BSF a preferred optimization criterion is included in the job request (in Fig. 1, criteria A, B and C). This criterion is used when allocating execution alternatives for the batch jobs in BS. Besides, the procedure of the initial job batch separation into a set of sub-batches and each sub-batch scheduling at the same given scheduling interval is introduced in BS. The job batch "slicing" increases the number of alternatives found for high-priority jobs, diversifies the choice on the slots combination selection stage, and therefore increases the resource sharing efficiency according to the VO policy.

Backfilling [7] responds to early resources releases and performs "on the fly" rescheduling which is very important when a user job runtime estimation differs significantly from the actual job execution time. However there are some limitations of backfilling for distributed computing: for example, it generally provides inefficient resource load by criteria differed from an average job start. Nevertheless, backfilling allows to load underutilized computing nodes with relatively low priority jobs. Thus, BSF combines capabilities of both BS and BF algorithms.

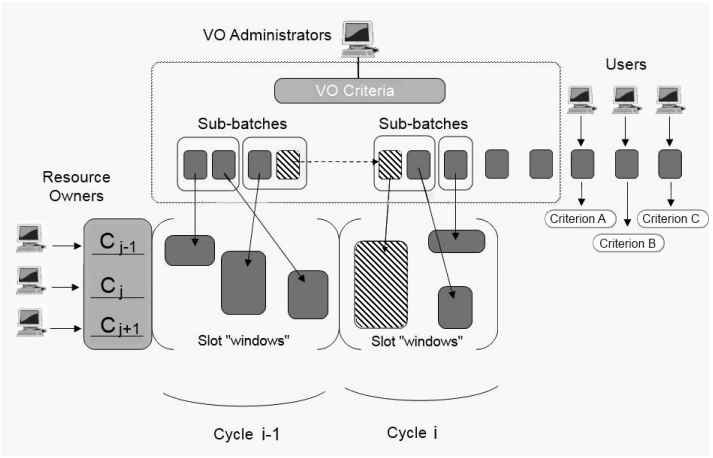


Fig. 1 Cyclic scheduling with BSF

The ratio between BS and BF scheduling may be set during the BSF sub-batches allocation.

As a next step of BSF generalization, a heuristic procedure of shifting the job execution alternatives (selected by BS for advanced reservation) is proposed. After BS scheduling the procedure shifts the alternatives in time towards the beginning of the scheduling interval keeping resource instances in which they are allocated. The shifting procedure is performed iteratively for each job of the batch being scheduled. The order of job selection is determined according to the start time of the chosen alternatives: first of all, an attempt to shift the alternatives with the minimal start time is performed. Such order guarantees that when shifting a job all other jobs with an earlier start time are already shifted and hence they do not occupy the corresponding nodes. Otherwise, a task with an earlier start time and a lower priority may block the shift of a task with a higher priority. Moreover, in its turn, this task may be shifted releasing extra slots that will not be utilized.

4 Simulation Studies

4.1 Scheduling Efficiency and Resource Level

The experiments consider scheduling efficiency studies using the proposed approaches (CSS, BSF, Shifted CSS), and also BF. The goal is to compare scheduling efficiency depending on the number of computational nodes in the domain as well as to investigate the consistency of schedules under conditions of inaccurate user job execution time estimations. A series of studies were carried out with the simulation environment [6]. Each experiment includes the generation of an input batch

with 15 jobs as well as the composition of resource structure and local schedules of a computational environment. To analyze the approaches under different conditions the simulation is conducted individually for different numbers of available nodes {6, 10, 20, 30, 50, 75, 100, 150}. Thus, the investigation is performed in terms of comparing the scheduling results obtained with the same input data by means of different algorithms.

Scheduling efficiency is considered from the viewpoint of a job batch total slot utilization time T_{proc} minimization, start t_{start} and finish job batch execution times minimization, and minimization of a combined criterion $F = t_{start} + T_{proc}$.

Figure 2 shows the average scheduled start times of the jobs obtained independently with all considered scheduling approaches depending on the computational environment nodes number.

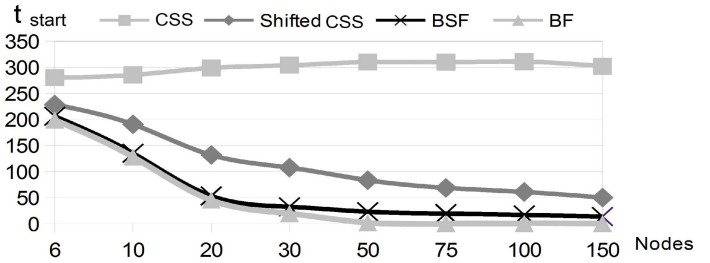


Fig. 2 Average job batch start time

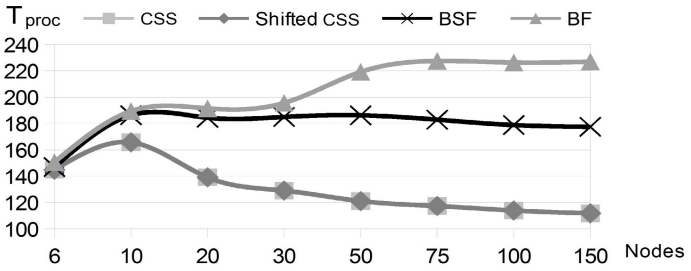


Fig. 3 Average batch jobs processor time usage

As one can see from Fig. 2, with an increasing amount of available computational nodes backfilling is able to reduce the average job batch start time down to zero (i.e. all batch jobs can start at the very beginning of the scheduling interval). At the same time the average jobs' start time obtained with CSS is almost independent from the number of the nodes available. BSF, as expected, provides the average job batch start time close to the one provided with backfilling by *filling* unused by CSS time slots near beginning of the scheduling interval with relatively low priority jobs. With a relatively large resource level an average job batch start time obtained with Shifted

CSS tends to a non-zero value since the most profitable in terms of the optimization criterion resources are generally allocated for more than one job. Thus in case of heterogeneous resource environment it is virtually impossible to start all the batch jobs at the beginning of the scheduling interval using CSS (even *shifted* variation).

Figure 3 shows CSS, Shifted CSS, and BSF advantage over backfilling on the VO target optimization criterion T_{proc}. It should be noted that with increasing of available resources number the advantage of CSS and BSF over backfilling also increases. The use of additional heuristics, such as job batch slicing, can provide even greater CSS advantage on the target criterion.

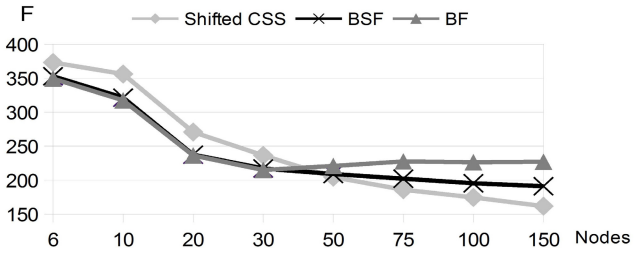


Fig. 4 Average batch jobs F value

Figure 4 shows the value of the combined resource usage efficiency index $F = t_{start} + T_{proc}$. It is important to note that the intersection between the Shifted CSS, BSF and backfilling graphs implies that in case of a relatively low level of available resources backfilling or BSF (they provide almost the same criterion F value) are more profitable in comparison to Shifted CSS. With the increasing computational environment size Shifted CSS becomes more advantageous in terms of resource usage efficiency. The same conclusion can be drawn if we evaluate the resource usage efficiency by the average batch jobs finish time: Shifted CSS provides the best results when a sufficiently large number of computational is nodes available. Thus the use of BSF is justified in virtually any conditions: this combined approach provides competitive to backfilling values of the considered resource usage efficiency indexes, and at the same time optimizes high-priority jobs execution performance.

4.2 Scheduling Efficiency Depending on Resource and Price Heterogeneity

For the hierarchical structure of job flow distribution (see Fig. 1) key problems include configuring and composing of computational resource domains and also choosing local scheduling algorithms at the level of individual domains. Two approaches to a computational resources composition can be defined: based on parameters similarity (performance, utilization, cost, etc.) or, for instance, based on geographical

location and maximum connection between resources. In the first case we deal with relatively homogeneous resources while in the second case the domains may have a relatively more heterogeneous resource composition (by performance, cost). Thus the challenge is to investigate the efficiency of the algorithms in question from the viewpoint of efficiency of the available domain resources utilization depending on the degree of computational environment heterogeneity. The following is the input data for the experiment: scheduling is conducted for a batch of 15 independent jobs in a domain consisting of 30 computational nodes. The behaviour of the algorithms in question in this configuration is already partly investigated and is presented in Fig. 2-4 (see Nodes = 30 by the X-axis). The experiment consists of several stages. At the first stage the domain is formed out of nodes with performance p in [5; 6] relative units. Thus we have a domain with a low level of resource heterogeneity. At the second stage the performance of domain nodes varies within the limits p in [2; 10] characterizing a medium level of resource heterogeneity. At the third stage of the experiment performance of domain resources varies within the limits p in [1; 20] (a high level of resource heterogeneity). Besides, at each of these stages two levels of pricing policy are considered: a conservative one when the price of resource use ('base' price) is a function of its performance (any two resources with the same performance will have the same price) and an aggressive one when the deviation from the "base" resource price is a random variable and may decrease (discount) and increase (extra charge) up to 60%. Note that the experiments in section 4.1 were conducted at a medium level of domain resource heterogeneity under the conditions of aggressive pricing policy.

Table 1 Average job batch start and processor time usage

Resource heterogeneity	Price policy	Start time			Processor time usage		
		BF	**BSF**	**CSS**	**BF**	**BSF**	**CSS**
Low	Conservative	19,8	24,4	61,7	164,5	164,4	162,5
	Aggressive	21	42,2	114,7	165,3	162,9	156,9
Medium	Conservative	18,5	38,7	111	158,9	158,2	142,9
	Aggressive	19,3	45,3	133,6	158	149,6	123,8
High	Conservative	11,1	27,4	77,3	104,8	104,7	76,7
	Aggressive	10,3	28	87	101,7	94,6	60

Table 2 shows values of the average start time and the average execution time of the jobs with low, medium and high level of resource heterogeneity at conservative and aggressive pricing policy. When environment utilization is low the minimal job start time is provided by backfilling (its advantage over cyclic scheduling scheme reaches 80%). On the other hand the advantage of the cyclic scheme against total scheduling time criterion is negligible (about 3%). This is mainly due to homogeneity of the batch jobs execution alternatives: under conditions of low domain resource heterogeneity the maximal difference in job execution in theory cannot exceed 15%. Aggressive pricing policy serves to decrease job execution time when using CSS and BSF but does not produce a significant effect. At the medium level of domain

resource heterogeneity backfilling retains advantage by average job start time, however, the CSS advantage over backfilling on processor time gets more considerable reaching 25% in case of aggressive pricing policy. When heterogeneity of domain resources is high the advantage of backfilling by job start time remains at the level of 80% while the advantage of CSS continues to increase and reaches 40%. Based on the experiment results we can conclude that in case of homogeneous domain resources the most efficient algorithm from the viewpoint of available resources is backfilling. However in case when the domain consists of resources with considerably varied performance it makes sense to use CSS or BSF. The cyclic scheduling scheme provides a better total execution time value, whereas BSF does not show the worst value by almost all considered efficiency indicators of domain resource usage compared to backfilling and at the same time guarantees efficient high priority jobs execution according to the specified CSS criterion.

5 Conclusions and Future Work

In this work, we compare the job batch scheduling results in terms of a virtual organization policy and the available resources usage efficiency. Based on a combination of a cyclic scheduling scheme (CSS or BS) and backfilling a BSF hybrid approach is proposed. Additionally the shifting procedure is proposed for the alternatives chosen in CSS. The simulation results show that depending on the considered scheduling efficiency index, and depending on the the resources availability, each of the considered approaches may provide the best results. Backfilling in general minimizes job start and finish times, while CSS is able, for example, to minimize the job processor time usage (when given the appropriate optimization criterion). In order to ensure the balanced scheduling results it is justified to use BSF: scheduling of high priority jobs with CSS and further filling the remaining underutilized resources with backfilling. The results obtained remain valid in a dynamically changing computational environment condition and topology, and in case when user jobs runtime estimations are significantly inaccurate. Further research will be related to an investigation of dividing the job flow into sub-batches depending on the jobs' requests and computational environment characteristics. Another research direction will be focused on CSS rescheduling based on the information about computational nodes current state and performance.

Acknowledgements. This work was partially supported by the Council on Grants of the President of the Russian Federation for State Support of Leading Scientific Schools (SS-362.2014.9) and the Russian Foundation for Basic Research (grant no. 12-07-00042).

References

1. Garg, S.K., Konugurthi, P., Buyya, R.: A Linear Programming-driven Genetic Algorithm for Meta-scheduling on Utility Grids. J. Par., Emergent and Distr. Systems 26, 493–517 (2011)
2. Lee, Y.C., Wang, C., Zomaya, A.Y., Zhou, B.B.: Profit-driven Scheduling for Cloud Services with Data Access Awareness. J. Par. and Distr. Computing 73(4), 591–602 (2012)
3. Kurowski, K., Nabrzyski, J., Oleksiak, A., Weglarz, J.: Multicriteria Aspects of Grid Resource Management. In: Nabrzyski, J., Schopf, J.M., Weglarz, J. (eds.) Grid Resource Management. State of the Art and Future Trends, pp. 271–293. Kluwer Academic Publishers, Boston (2003)
4. Rodero, I., Villegas, D., Bobroff, N., Liu, Y., Fong, L., Sadjadi, S.M.: Enabling Interoperability among Grid Meta-Schedulers. J. Grid Computing 11(2), 311–336 (2013)
5. Toporkov, V., Toporkova, A., Tselishchev, A., Yemelyanov, D., Potekhin, P.: Metascheduling and Heuristic Co-allocation Strategies in Distributed Computing. Computing and Informatics 6 (to be published , 2014)
6. Toporkov, V., Tselishchev, A., Yemelyanov, D., Bobchenkov, A.: Composite Scheduling Strategies in Distributed Computing with Non-dedicated Resources. Procedia Computer Science 9, 176–185 (2012)
7. Moab Adaptive Computing Suite, `http://www.adaptivecomputing.com/products/moab-adaptive-computing-suite.php`
8. Ernemann, C., Hamscher, V., Yahyapour, R.: Economic Scheduling in Grid Computing. In: Feitelson, D.G., Rudolph, L., Schwiegelshohn, U. (eds.) JSSPP 2002. LNCS, vol. 2537, pp. 128–152. Springer, Heidelberg (2002)
9. Cafaro, M., Mirto, M., Aloisio, G.: Preference-Based Matchmaking of Grid Resources with CP-Nets. J. Grid Computing 11(2), 211–237 (2013)
10. Aida, K., Casanova, H.: Scheduling Mixed-parallel Applications with Advance Reservations. In: 17th IEEE Int. Symposium on HPDC, pp. 65–74. IEEE CS Press, New York (2008)
11. Ando, S., Aida, K.: Evaluation of Scheduling Algorithms for Advance Reservations. Information Processing Society of Japan SIG Notes. HPC-113, 37–42 (2007)
12. Elmroth, E., Tordsson, J.: A Standards-based Grid Resource Brokering Service Supporting Advance Reservations, Coallocation and Cross-Grid Interoperability. J. of Concurrency and Computation 25(18), 2298–2335 (2009)
13. Toporkov, V., Toporkova, A., Tselishchev, A., Yemelyanov, D.: Slot Selection Algorithms in Distributed Computing with Non-dedicated and Heterogeneous Resources. In: Malyshkin, V. (ed.) PaCT 2013. LNCS, vol. 7979, pp. 120–134. Springer, Heidelberg (2013)
14. Azzedin, F., Maheswaran, M., Arnason, N.: A Synchronous Co-allocation Mechanism for Grid Computing Systems. Cluster Computing 7, 39–49 (2004)
15. Castillo, C., Rouskas, G.N., Harfoush, K.: Resource Co-allocation for Large-scale Distributed Environments. In: 18th ACM International Symposium on High Performance Distributed Compuing, pp. 137–150. ACM, New York (2009)
16. Takefusa, A., Nakada, H., Kudoh, T., Tanaka, Y.: An Advance Reservation-based Co-allocation Algorithm for Distributed Computers and Network Bandwidth on QoS-guaranteed Grids. In: Frachtenberg, E., Schwiegelshohn, U. (eds.) JSSPP 2010. LNCS, vol. 6253, pp. 16–34. Springer, Heidelberg (2010)
17. Blanco, H., Guirado, F., Lérida, J.L., Albornoz, V.M.: MIP Model Scheduling for Multi-Clusters. In: Caragiannis, I., Alexander, M., Badia, R.M., Cannataro, M., Costan, A., Danelutto, M., Desprez, F., Krammer, B., Sahuquillo, J., Scott, S.L., Weidendorfer, J. (eds.) Euro-Par Workshops 2012. LNCS, vol. 7640, pp. 196–206. Springer, Heidelberg (2013)
18. Olteanu, A., Pop, F., Dobre, C., Cristea, V.: A Dynamic Rescheduling Algorithm for Resource Management in Large Scale Dependable Distributed Systems. Computers and Mathematics with Applications 63(9), 1409–1423 (2012)
19. Buyya, R., Abramson, D., Giddy, J.: Economic Models for Resource Management and Scheduling in Grid Computing. J. Concurrency and Computation 14(5), 1507–1542 (2002)

Locality Aware Task Scheduling in Parallel Data Stream Processing

Zbyněk Falt, Martin Kruliš, David Bednárek, Jakub Yaghob, and Filip Zavoral

Abstract. Parallel data processing and parallel streaming systems become quite popular. They are employed in various domains such as real-time signal processing, OLAP database systems, or high performance data extraction. One of the key components of these systems is the task scheduler which plans and executes tasks spawned by the system on available CPU cores. The multiprocessor systems and CPU architecture of the day become quite complex, which makes the task scheduling a challenging problem. In this paper, we propose a novel task scheduling strategy for parallel data stream systems, that reflects many technical issues of the current hardware. We were able to achieve up to $3\times$ speed up on a NUMA system and up to 10% speed up on an older SMP system with respect to the unoptimized version of the scheduler. The basic ideas implemented in our scheduler may be adopted for task schedulers that focus on other priorities or employ different constraints.

Keywords: Parallel, multicore CPU, NUMA, cache aware, task scheduling, data streams.

1 Introduction

Parallel processing is becoming increasingly important in high performance systems, since the hardware architectures have embraced concurrent execution to increase their computational power. Unfortunately, parallel programming is much more difficult and error prone, since the programmers are used to think and express their intentions in serial manner. Many different paradigms and concepts have been devised to simplify the design of concurrent processing.

Zbyněk Falt · Martin Kruliš · David Bednárek · Jakub Yaghob · Filip Zavoral
Parallel Architectures/Applications/Algorithms Research Group,
Faculty of Mathematics and Physics,
Charles University in Prague, Czech Republic
e-mail: {falt,krulis,bednarek,yaghob,zavoral}@ksi.mff.cuni.cz

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_35

One of these approaches is stream data processing. It was originally designed for systems that process data which are generated in real-time and need to be processed immediately, but they also have been adopted in database systems and parallel systems, since they simplify the application design and naturally reveal opportunities for parallelization.

A streaming application is usually expressed as an oriented graph (also denoted as *execution plan*), where the vertices are processing stages (operators or filters) that process the data and the edges prescribe how the data are passed on between these stages. The main advantage from the perspective of parallel processing is that the each stage contains serial code (which is easy to design) and multiple stages may be executed concurrently.

One of the systems that implements this idea is Bobox [5]. The main objective of Bobox is to process semantic and semi-structured data effectively [6]. It currently supports the SPARQL [18] query language and partially the XQuery [8] and the TriQuery [4] language. One of the most challenging problems of this system is to effectively and efficiently execute the work of the operators on the available CPU cores.

In this paper, we propose a novel locality aware task scheduling strategy (called LAS) for data streaming systems. This strategy incorporates important hardware factors such as cache hierarchies and non-uniform memory architectures (NUMA). We have implemented this strategy in the Bobox task scheduler and achieved significant speedup on modern host systems. Although our performance analysis was conducted using Bobox, the scheduler itself can be easily adopted for other streaming systems as well.

The paper is organized as follows. Section 2 revise the most important facts regarding state-of-the-art CPU architectures and NUMA systems. Our LAS scheduler is described in Section 3. Section 4 presents the experimental results that evaluate the benefits of our innovations. The related work is revised in Section 5 and Section 5 concludes the paper.

2 CPU Fundamentals and Task Scheduling

In this section, we revise fundamental facts regarding the architectures of modern multi-core CPUs and NUMA systems. We also put these facts in the perspective of task scheduling which is often employed to achieve parallel data processing in complex systems.

2.1 CPU Architecture

The CPU architectures became quite complex in the past few decades. We will focus solely on the properties, which directly affect the parallel execution of tasks that cooperate via shared memory. A generic schema of modern multi-core CPU is presented in Figure 1.

The CPU comprises several physical cores which are quite independent. These cores usually share only the memory controller and sometimes certain levels of cache.

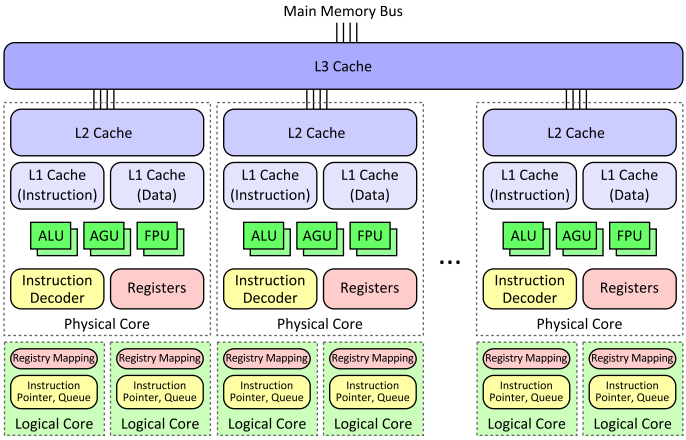


Fig. 1 A schema of multi-core CPU

The physical cores are often divided into two logical cores by means of hyper threading (Intel) or dual-core modules (AMD) technology. The logical cores share also some computational units and also the lowest level of cache. In the remainder of the paper, we will use the term CPU core to denote logical cores – i.e., the lowest computational unit of the CPU which processes one thread at a time.

Multiprocessor configurations combine several CPUs into one system. Older CPUs, which does not have integrated memory controllers are connected in a similar way as the physical cores in the CPU die. This configuration is called symmetric multiprocessing (SMP). Newer CPUs have memory controllers integrated, thus each CPU manages part of the system memory. This configuration is called non-uniform memory architecture (NUMA), since the memory latency depends on whether it is directly connected to the CPU who uses it, or whether it needs to be accessed via a controller of another CPU.

The organization of the cores within the CPU and organization of the CPU chips within a NUMA system forms a hierarchy. Logical cores of one physical core or physical cores that share memory cache are considered close, while cores on two different CPU chips (i.e., in two different NUMA nodes) are considered distant. This hierarchy plays significant role in task planning. Related or cooperating tasks should be scheduled on cores that are close by, since they will likely benefit from cache sharing. Completely unrelated tasks using different portions of the main memory should be scheduled on different NUMA nodes, so they can keep their intermediate data in different caches and in different memory nodes.

2.2 Task Scheduling

In order to achieve parallelism on modern CPUs, the work needs to be divided into portions that can be processed concurrently. Traditional division into threads is too coarse and tedious, hence most of the parallel systems deal with tasks. The *task* comprises both the data and the procedure that process the data. Tasks are scheduled and processed by available CPU cores. It has been established [19] that the tasks can be effectively employed in the implementation of more complex parallel patterns such as parallel loops, reduction, pipeline, or data stream processing.

In this work, we focus solely on systems where the tasks are generated dynamically by other tasks or by external events (e.g., user requests). Such systems must employ dynamic scheduling, which can cope with the ever changing situation. The dynamic scheduler manages the tasks which are ready to run and when to assign them to the available CPU cores as they become available.

Furthermore, task schedulers often employ some form of restrictions for implicit synchronization like task dependencies. When a task is spawned, it may not be ready to execute immediately. In such case, the task scheduler needs to manage *waiting* tasks along with the *ready* tasks. When a waiting task conditions for execution are met, the scheduler change its state to *ready* and eventually assigns it to an available CPU core. However, we are focusing on improving efficiency of the task scheduling, thus we will not consider the waiting tasks nor any mechanisms for automatic testing the task readiness. Henceforth, we use the term *task spawning* for introducing a ready tasks to the scheduler.

3 Locality Aware Task Scheduler

The task scheduler manages tasks in the system and process them on the available computational units. Different task schedulers may be used for different systems. In our work, we address the problems of parallel data processing, such as problems of database management systems. Hence, our objective is to design a task scheduler that reflects three important issues:

- Even though the overall work is orchestrated by some form of an execution plan, the interpretation of the plan is data dependent, thus the tasks are spawned dynamically.
- The tasks should be planed with respect to overall throughput of the system, since they usually work on a complex problem which needs to be solved as whole.
- The available hardware resources should be utilized efficiently.

The last issue is becoming increasingly important as the CPU architectures are getting more complex with each new generation. Planning the tasks in a way that considers which data are hot in caches or that better organizes the work among NUMA nodes is the key to achieving much better overall performance.

In order to achieve better results, we have improved the definition of a task, so the programmer of the system that employs our scheduler can pass on some explicit information which can be used for scheduling. First of all, we distinguish two types of tasks [7] and this type is specified when the task is spawned:

- The *immediate task* represent work that immediately relates to the task being currently processed. This type of tasks is expected to be executed as soon as possible and preferably close to the task that spawned them to utilize data which are still hot in the cache.
- The *deferred task* represent work that is not closely related to the task being currently processed. This type of tasks is also expected to generate more sub tasks eventually.

Furthermore, every task (both immediate and deferred) is attached to a *request* which corresponds to a larger portion of work that is divided into tasks to achieve parallelism (e.g., it can be related to a database query). Requests are uniquely identified by a *request ID*, which is a sequentially assigned number. A task inherits its request ID from the task which spawned it.

3.1 Task Scheduling Strategy

The initialization process of the task scheduler scans the host system and detects the configuration and properties of the CPUs. CPU cores which share at least one level of cache are bundled together in logical *core groups* and a *thread pool* is created for each group. The thread pool has one thread for each CPU core in the corresponding group and the threads have their affinity set to this core group. The thread can easily determine the associated CPU core using appropriate operating system functions.

Each core group maintains one queue of immediate tasks per core and one shared queue of deferred tasks. The deferred queue is in fact more complex data structure than a simple queue, which maintains the task of each request separately. It also provides quick access and extraction of the youngest and the oldest tasks from the oldest and second oldest request.

The main paradigm employed in the LAS scheduling strategy is to emphasize data locality awareness. Therefore, when the scheduler assigns another work to a thread, it attempts to select a task which is as close as possible to the previous work done by that thread. For this purpose, we define the distance between two cores within one group and the distance between two core groups. Distance of cores within one group is equal to the lowest level of cache these two cores are sharing. Distance of two core groups is equal to the distance of their corresponding NUMA nodes (and zero for groups that share a NUMA node).

The distance between cores and core groups determine our scheduling strategy. When a thread completes a task it executes the scheduling algorithm to fetch another task to execute. The first applicable rule of the following list is taken:

1. The youngest task from the queue of immediate tasks of the current core.
2. Other cores of the same group are scanned (in the increasing distance) and the first non-empty immediate queue is found. If such queue exists, its oldest task is taken.
3. The youngest deferred task of the oldest request from the deferred task queue of the current group is taken. This rule ensures that all threads of one core group work on the same (the oldest) request if possible.
4. Other core groups are scanned (in increasing distance) and the first non-empty deferred queue is found. If such queue exists, the oldest deferred task of the **second** oldest request is taken. If the queue has tasks of only one request, its oldest task is taken instead. This strategy assumes that a core group is heavily engaged in the processing of the oldest request and it would not be wise to disrupt this work when another request is available. However, this algorithm does not prevent the situation that the whole system cooperates on one common request.
5. Immediate queues of cores from other groups which are located on the same NUMA node[1] are scanned. If non-empty queue is found, its oldest task is taken. The immediate queues are scanned in round robin manner and the thread remembers the last non-empty queue found. When this rule is applied again, the scan is resumed where it previously ended. This rule enforces that all immediate tasks are processed on the same NUMA node where they were spawned.

If all steps fail (i.e., there is no available task to execute), the thread is suspended, so it will not consume system resources. The whole algorithm and the thread group hierarchy is depicted in Figure 2.

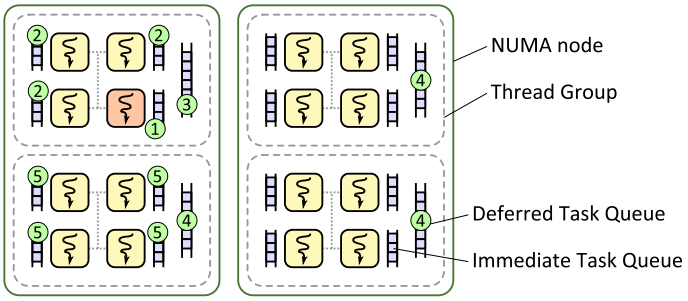


Fig. 2 The schema of a task scheduler

When a thread spawns a tasks, it first determines on which CPU core it runs. The immediate tasks are inserted to the immediate queue of the core. The deferred tasks are inserted in the shared queue of deferred tasks of the corresponding core group.

[1] Note that no such group may exist when exactly one group is assigned to each NUMA node.

3.2 Resuming Suspended Threads

When a thread does not have another task to process, the thread suspends itself on a synchronization primitive[2]. Each group has one such synchronization primitive and the suspended threads are added to its waiting queue.

When a new task is spawned, the spawning thread attempts to wake one of the suspended threads. First, it tries to wake a thread in the same thread pool (the same group of cores). If the whole group is working, it scans all other groups and attempts to wake a thread there. The groups are scanned in an increasing distance from the original group and the search finishes when a group with suspended thread is found or all groups are scanned.

The immediate tasks and deferred tasks are handled slightly differently in this case. When an immediate task is spawned, the search for a group with suspended thread ends at the boundary of the NUMA node. There is no need to wake threads on other NUMA nodes since the scheduling rules prevent the immediate tasks to travel between NUMA nodes. When deferred tasks is spawned, all groups are tested.

4 Experiments

We performed several experiments to prove that the LAS scheduling algorithm significantly improved performance of the system. For the testing, we used our parallel implementation of the in-memory SPARQL engine [14] and the SP^{2}Bench benchmark [21] and its 5m testing dataset. The implementation of the engine is able to generate parallel execution plans without significant serial bottlenecks, i.e., all worker threads are utilized during their evaluation. Additionally, the SP^{2}Bench benchmark contains several queries which generate various and really complex query execution plans. This is profitable since this variety shows the behaviour of the task scheduler under various circumstances.

We selected queries Q2, Q4, Q5a, Q6, Q7 Q8, Q9 and Q11 from the benchmark, since they take reasonable time to evaluate and their query execution plans are complex enough. Other queries are evaluated so fast that the results are negligible.

We used two hardware configurations for the experiments:

- A server with two Intel Xeon E5310 processors, both running at 1.60Ghz. This type of processor has 4 cores and two shared 4MB L2 caches. First two cores share the first L2 cache, second two cores share the other. Additionally, each core has its own L1 cache (32kB + 32kB). This configuration represents non-trivial SMP system and our scheduling strategy creates 4 thread pools for this configuration.
- A NUMA server with four Intel Xeon E7-4820 processors, all running at 2.0Ghz. This type of processor has physical 8 cores with Hyper-Threading Technology, i.e., the processor has 16 logical cores in total. Each physical core has its own

[2] Current implementation uses standard semaphore and atomic operations that handles related metadata.

L1 cache (32kB + 32kB), L2 cache (256kB) and all cores share one L3 cache (18MB). This configuration represents non-trivial NUMA system and our scheduling strategy creates 4 thread pools as well.

We performed two different experiments for each hardware configuration:

- We run the selected query just once. This experiment demonstrate the situation when there is a lot of various data dependencies among the tasks, since all tasks belong to one query.
- We run the selected query multiple times in parallel (16 times in all measurements). This experiment demonstrate the situation when there is a lot of tasks which do not have any dependency on each other, i.e., each thread can process its own instance of the query without any cooperation with the others.

Finally, we performed two different measurements for each experiment:

- We used the scheduler which implements the LAS scheduling strategy described in the Section 3.
- We used the scheduler which implements the strategy which is close to the strategy used in TBB or in our previous work [7]. Each thread keeps its local queue of immediate task and all threads share one queue of deferred tasks. Thread executes the first existing task in this order: the latest immediate task from its local queue, the oldest deferred task from the shared queue and the oldest immediate task from the local queue of another thread. In other words, immediate tasks are handled in the same manner as the spawned tasks in TBB [2] and deferred tasks are handled in the same manner as the enqueued tasks in TBB. We denote this scheduler as *NLS*.

4.1 SMP System

The results are shown in Figure 3 for single query and in Figure 4 for multiple parallel queries. Notice that for Q6 only 1m dataset was used so that the evaluation takes reasonable time. In multiple queries, we used 1m dataset for Q4 and Q8 in order to avoid swapping of the operating memory. Additionally, we used only 250k dataset for Q6 from the same reason as in the single query.

Single Query

As expected, on SMP system the NLS scheduling algorithm performs quite well. However, queries Q4 and Q6 benefits from the LAS and especially query Q2 is almost $2\times$ faster. This query consist of one long pipeline, therefore, it is especially sensitive to the data locality. Other queries contains such pipelines as well, however, these pipelines are typically split to multiple independent parts because of sorting

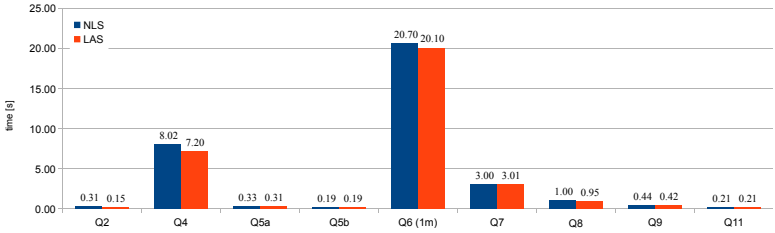


Fig. 3 Single query on SMP

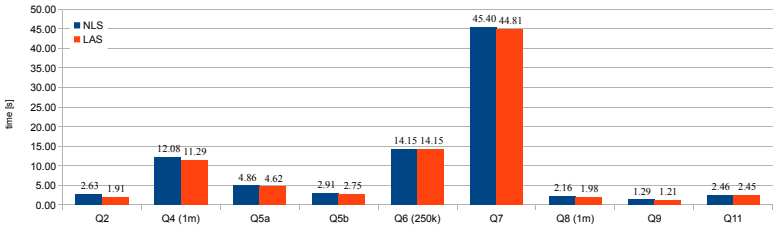


Fig. 4 Multiple queries on SMP

operators which break up the pipeline processing and cause that the execution plans are evaluated by parts, i.e., that all threads cooperate close to each other.

Multiple Queries

The second experiment shows that the LAS performs better than NLS and in more cases than in the first experiment. The main reason is that the LAS better separates individual requests, i.e., the requests do not force out each other from cache memory.

4.2 NUMA System

Both experiments on the NUMA system (see Figure 5 and Figure 6) proves that taking the NUMA factor into account is very important in modern systems. The main problem is that accessing memory of another node slows down both communicating nodes and the system bus.

The LAS tries to keep one request on one NUMA node as much as it is possible. If it is not possible, it tries to keep different branches of execution plans on different NUMA nodes which minimizes data interference between the NUMA nodes. The NLS does not distinguish among the NUMA nodes, therefore, the relationship between a thread and the memory being accessed is almost arbitrarily. This is significant especially in the multiple queries.

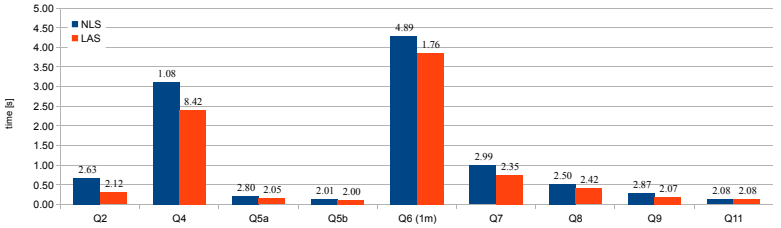


Fig. 5 Single query on NUMA

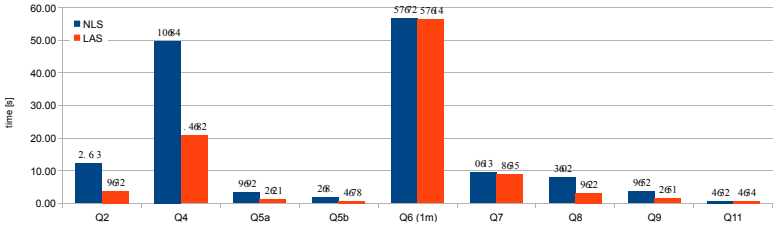


Fig. 6 Multiple queries on NUMA

All performed experiments proved that the locality awareness has a significant impact on the performance of the parallel streaming system running on either SMP or NUMA systems.

5 Related Work

As we already mentioned in Section 2, modern architectures became complex and complicated. Thus, optimizing the performance of applications through elaborate task scheduling strategies is a challenging task and a very hot topic in current research. The fact that finding an optimal scheduling plan is NP-hard problem causes that all scheduling strategies just try to find a suboptimal solution using heuristics and approximation techniques [22].

In streaming systems, there are several aspects of scheduling optimization, such as memory usage [3], cache-efficiency [12], response time, throughput, etc. or their mutual combinations [15, 20].

In this work, we relaxed many aspects and just tried to maximize data locality in order to increase the performance of the system. This allowed us to adopt techniques used in non-streaming systems. Our previous work [7] was the first step. We showed that data flow awareness (i.e., using immediate and deferred tasks) in streaming systems increase data locality; however, the scheduling algorithm lacks the support of NUMA and non-trivial SMP systems.

These issues are successfully solved in several works, e.g., in popular parallel frameworks such as OpenMP [9, 13] or Intel Threading Building Blocks [2, 17].

The division of tasks to immediate and deferred tasks ensures that threads work with data hot in cache if they have its own immediate tasks. However, it showed up,

that the bottleneck of the system is the task stealing since the tasks were stolen from a randomly chosen thread. However, this is an issue of the cited papers as well.

The task stealing optimization is researched thoroughly in work by Chen et. al. [10, 11]. In fact, the algorithm CATS/CAB from this work is similar to our LAS algorithm described in Section 3; however, there are several differences between these two algorithms. LAS partitions physical processors more precisely according to the structure of shared caches, whilst CATS/CAB creates always one group per physical processor (socket). Furthermore, LAS algorithm for task stealing within a group also considers the cache hierarchy, which is beneficial when the cores in one group share more than last level of cache. Additionally, the LAS sets affinity of threads together for the whole group. This has two advantages – first, we can freely add and remove threads to the thread pool which enables support of IO operations [16], second, this strategy copes better with Hyper-Threading Technology, since it does not restrict the operating system from its own load balancing strategy [1]. Finally, we optimize the situation when the system processes multiple independent requests.

6 Conclusions

In this paper, we presented a novel task scheduling strategy that takes advantages on current CPU architectures and both SMP and NUMA multiprocessor systems. Our scheduler can effectively improve the data locality and thus the cache reusability when employed on parallel data stream processing systems. We have implemented a prototype of the scheduler and integrated it into the Bobox framework, which allows creation and evaluation of the execution plans. When applied on a SPARQL benchmark that process RDF data, the system achieved up to 10% speed up on double-processor SMP system and up to $3\times$ speed up on four processor NUMA system for selected queries with respect to previous version of the scheduler.

In the future work, we would like to extend our scheduler to other domains of task processing. We would like to improve generic frameworks that also use tasks to achieve parallelism, but which process different types of datasets (not only streaming data). Furthermore, we would like to extend the scheduler to support work offloading to parallel accelerators such as GPUs and Xeon Phi cards, where the data transfers between the host system and the parallel device need to be considered.

Acknowledgements. This work was supported by the Czech Science Foundation (GACR), projects P103-13-08195S and P103-14-14292P, and by Specific Research project SVV-2014-260100.

References

1. Impact of Load Imbalance on Processors with Hyper-Threading Technology (2011), `http://software.intel.com/en-us/articles/impact-of-load-imbalance-on-processors-with-hyper-threading-technology` (accessed March 18, 2014)

2. Intel Threading Building Blocks Reference Manual (2014), http://software.intel.com/en-us/node/506130 (accessed March 18, 2014)
3. Babcock, B., Babu, S., Datar, M., Motwani, R., Thomas, D.: Operator scheduling in data stream systems. The International Journal on Very Large Data Bases 13(4), 333–353 (2004)
4. Bednárek, D., Dokulil, J.: TriQuery: Modifying XQuery for RDF and Relational Data. In: 2010 Workshops on Database and Expert Systems Applications, pp. 342–346. IEEE (2010)
5. Bednárek, D., Dokulil, J., Yaghob, J., Zavoral, F.: The Bobox Project - A Parallel Native Repository for Semi-structured Data and the Semantic Web. In: ITAT - IX. Informačné technológie - aplikácie a teória, pp. 44–59 (2009)
6. Bednárek, D., Dokulil, J., Yaghob, J., Zavoral, F.: Using methods of parallel semi-structured data processing for semantic web. In: International Conference on Advances in Semantic Processing, pp. 44–49 (2009)
7. Bednárek, D., Dokulil, J., Yaghob, J., Zavoral, F.: Data-flow awareness in parallel data processing. In: Fortino, G., Badica, C., Malgeri, M., Unland, R. (eds.) IDC 2012. SCI, vol. 446, pp. 149–154. Springer, Heidelberg (2012)
8. Boag, S., Chamberlin, D., Fernández, M., Florescu, D., Robie, J., Siméon, J., Stefanescu, M.: XQuery 1.0: An XML query language. W3C working draft 15 (2002)
9. Broquedis, F., Furmento, N., Goglin, B., Namyst, R., Wacrenier, P.-A.: Dynamic task and data placement over NUMA architectures: An openMP runtime perspective. In: Müller, M.S., de Supinski, B.R., Chapman, B.M. (eds.) IWOMP 2009. LNCS, vol. 5568, pp. 79–92. Springer, Heidelberg (2009)
10. Chen, Q., Guo, M., Huang, Z.: CATS: Cache Aware Task-stealing Based on Online Profiling in Multi-socket Multi-core Architectures. In: Proceedings of the 26th ACM International Conference on Supercomputing, ICS 2012, pp. 163–172. ACM, New York (2012)
11. Chen, Q., Huang, Z., Guo, M., Zhou, J.: Cab: Cache aware bi-tier task-stealing in multi-socket multi-core architecture. In: 2011 International Conference on Parallel Processing (ICPP), pp. 722–732. IEEE (2011)
12. Cieslewicz, J., Mee, W., Ross, K.: Cache-conscious buffering for database operators with state. In: Proceedings of the Fifth International Workshop on Data Management on New Hardware, pp. 43–51. ACM (2009)
13. Duran, A., Corbalán, J., Ayguadé, E.: Evaluation of openMP task scheduling strategies. In: Eigenmann, R., de Supinski, B.R. (eds.) IWOMP 2008. LNCS, vol. 5004, pp. 100–110. Springer, Heidelberg (2008)
14. Falt, Z., Čermak, M., Zavoral, F.: Highly Scalable Sort-Merge Join Algorithm for RDF Querying. In: Proceedings of the 2nd International Conference on Data Management Technologies and Applications (2013)
15. Jiang, Q., Chakravarthy, S.: Scheduling strategies for processing continuous queries over streams. In: Williams, H., MacKinnon, L.M. (eds.) BNCOD 2004. LNCS, vol. 3112, pp. 16–30. Springer, Heidelberg (2004)
16. Kruliš, M., Falt, Z., Bednárek, D., Yaghob, J.: Task scheduling in hybrid CPU-GPU systems. Informačné Technológie-Aplikácie a Teória, p. 17
17. Kukanov, A., Voss, M.: The foundations for scalable multi-core software in Intel Threading Building Blocks. Intel Technology Journal 11(4), 309–322 (2007)
18. Prud'Hommeaux, E., Seaborne, A., et al.: SPARQL query language for RDF. W3C working draft, 4 (2006)
19. Reinders, J.: Intel Threading building blocks. O'Reilly (2007)
20. Safaei, A.A., Haghjoo, M.S.: Parallel processing of continuous queries over data streams. Distrib. Parallel Databases 28, 93–118 (2010)
21. Schmidt, M., Hornung, T., Lausen, G., Pinkel, C.: SP^2Bench: a SPARQL performance benchmark. In: IEEE 25th International Conference on Data Engineering, ICDE 2009, pp. 222–233. IEEE (2009)
22. Sinnen, O.: Task scheduling for parallel systems, vol. 60. John Wiley & Sons (2007)

Part XI
Social Computing

A Survey of Social Web Mining Applications for Disease Outbreak Detection

Gema Bello-Orgaz, Julio Hernandez-Castro, and David Camacho

Abstract. Social Web Media is one of the most important sources of big data to extract and acquire new knowledge. Social Networks have become an important environment where users provide information of their preferences and relationships. This information can be used to measure the influence of ideas and the society opinions in real time, being very useful on several fields and research areas such as marketing campaigns, financial prediction or public healthcare among others. Recently, the research on artificial intelligence techniques applied to develop technologies allowing monitoring web data sources for detecting public health events has emerged as a new relevant discipline called Epidemic Intelligence. Epidemic Intelligence Systems are nowadays widely used by public health organizations like monitoring mechanisms for early detection of disease outbreaks to reduce the impact of epidemics. This paper presents a survey on current data mining applications and web systems based on web data for public healthcare over the last years. It tries to take special attention to machine learning and data mining techniques and how they have been applied to these web data to extract collective knowledge from Twitter

1 Introduction

Web is one of the most important sources of data in the world producing amounts of public information. The exponentially increasing of websites and online web

Gema Bello-Orgaz
Escuela Politecnica Superior, Universidad Autonoma de Madrid,
Francisco Tomas y Valiente 11, 28049 Madrid, Spain
e-mail: gema.bello@uam.es

Julio Hernandez-Castro
School of Computing, University of Kent, Cornwallis South Building, Canterbury CT2 7NF, UK
e-mail: J.C.Hernandez-Castro@kent.ac.uk

David Camacho
Escuela Politecnica Superior, Universidad Autonoma de Madrid,
Francisco Tomas y Valiente 11, 28049 Madrid, Spain
e-mail: david.camacho@uam.es

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_36

services in the last years has allowed new interdisciplinary challenges for several fields and computer science, such as marketing campaigns [8] [3], financial prediction [2] or public healthcare [10] [17] [7], among others. Recently, the research on artificial intelligence techniques applied to develop technologies allowing monitoring web data sources for detecting public health events has been emerged as a new relevant discipline called Epidemic Intelligence (EI).

EI can be defined as the early identification, assessment and verification of potential public health risks [25], and the timely dissemination of the appropriate alerts. This discipline includes surveillance techniques such as automated and continuous analysis of unstructured free text information available on Web from social networks, blogs, digital news media or official sources. Surveillance systems are nowadays widely used by public health organizations such as World Health Organization (WHO) or the European Centre for Disease Prevention and Control (ECDC) [16]. Tracking and monitoring mechanisms for early detection are critical in reducing the impact of epidemics giving a rapid response. For instance, several of these systems can be able to discover early events of the disease breakout during the A(H1N1) influenza pandemic in 2009 [11].

Traditional epidemic surveillance systems are implemented from virology and clinical data, which is manually collected, and often these traditional systems have a delay reporting the emerging diseases. But in situations like epidemic outbreaks, real-time feedback and a rapid response is critical. Social Web media is a profitable medium to extract the society opinion in real time. Blogs, micro-blogs (Twitter), and social networks (Facebook) enable people to publish their personal opinions in real time, including geo-information about their current locations. These big data with situation and context aware information about the users provide a useful source for public healthcare. However, the extraction of information from web is a difficult task due to its unstructured definition, high heterogeneity, and dynamically changing nature. Because of this diversity in the data format, several computational methods are required for its processing and analysing [17] (data mining, natural language processes (NLP), knowledge extraction, context awareness, etc...).

This paper presents a survey on current data mining applications and web systems based on web data for public healthcare over the last years. It tries to take special attention to machine learning and data mining techniques, and how they have been applied to these web data to extract collective knowledge from social networks like Twitter. The rest of the paper has been structured as follows: Section 2 shows the state of the art of the existing Epidemic Intelligence Systems. Section 3 describes the different web mining techniques used to detect disease outbreaks. Section 4 provides an overview of Twitter applications for monitoring and predicting epidemic and their experimental results. Finally, the last section presents a discussion of the main features extracted from this survey.

2 Epidemic Intelligence Systems for Public Healthcare

Nowadays, large amounts of emergency and health data are increasingly coming from a large range of web and social media sources. This information can be very useful for disease surveillance and early outbreak detection, and several public web surveillance projects in this field have emerged over the recent years.

One of the earliest surveillance systems is the Global Public Health Intelligence Network (GPHIN) [24] developed by Public Health Agency of Canada in collaboration with WHO. It is a secure web-based multilingual warning tool that is continuously monitoring and analysing global media data sources to identify information about disease outbreaks and other events related to public healthcare. The information is filtered for relevancy by an automated process, and categorized based on a specific taxonomy of categories. Then this information is analysed by Public Health Agency of Canada GPHIN officials. From 2002 to 2003 years, this surveillance system was able to detect the outbreak of Severe Acute Respiratory Syndrome (SARS).

Since 2006, BioCaster [11] is an operational ontology-based system for monitoring online media data. This system is based on text mining techniques for detecting and tracking the infectious disease outbreaks through the search of linguistic signals. The system continuously analyses documents reported from over 1700 RSS feeds, Google News, WHO, ProMED-mail, and the European Media Monitor, among others providers. The extracted text are classified for topical relevance and plots them onto a Google map using geo-information. The system consists of four main stages: topic classification, named entity recognition(NER), disease/location detection and event recognition. In the first stage, the texts are classified into relevant or non-relevant categories using to train a naive Bayes classifier. Then, for relevant document corpus are search entities of interest from 18 concept types based on the BioCaster ontology [12] related to diseases, viruses, bacteria, locations and symptoms, see Figure 1.

HealthMap project [5] is a global disease alert map which uses data from different sources such as Google News, expert-curated discussion such as ProMED-mail, and official organization reports such as World Health Organization (WHO) or Euro Surveillance. This is an automated real-time system that monitors, organizes, integrates, filters, visualizes, and disseminates online information about emerging diseases as can be seen in Figure 2.

Other system which collects news from the Web, related to human and animal health, and plot the data on a Google Maps mashupare is EpiSpider [18]. This tool automatically extracts infectious disease outbreak information from several sources including ProMed-mail and medical web sites, and it is used as surveillance system by public healthcare organizations, several universities and health research organization. Additonally, this system automatically converted the topic and location information of the reports into RSS feeds.

Other public health surveillance system used by a Public Health Organization (The European Centre of Disease Prevention and Control) is MedISys [23] monitoring human and animal infectious diseases, as well as chemical, biological, radiological and nuclear (CBRN) threats in open-source media. MedISys automatically

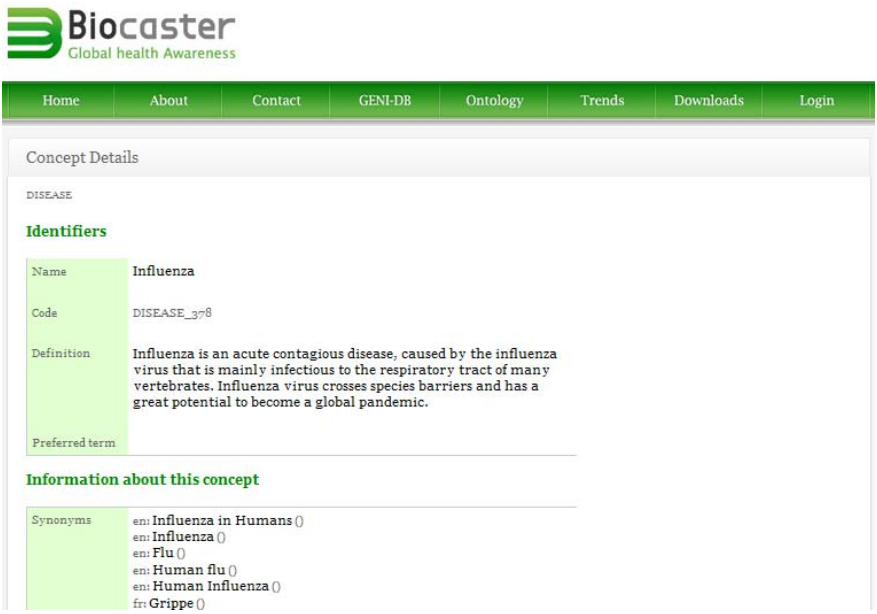


Fig. 1 BioCaster ontology related to diseases, viruses, bacteria, locations and symptoms. Screenshot taken from the BioCaster Health Monitor Web (http://born.nii.ac.jp), online accessed on 18th March 2014.

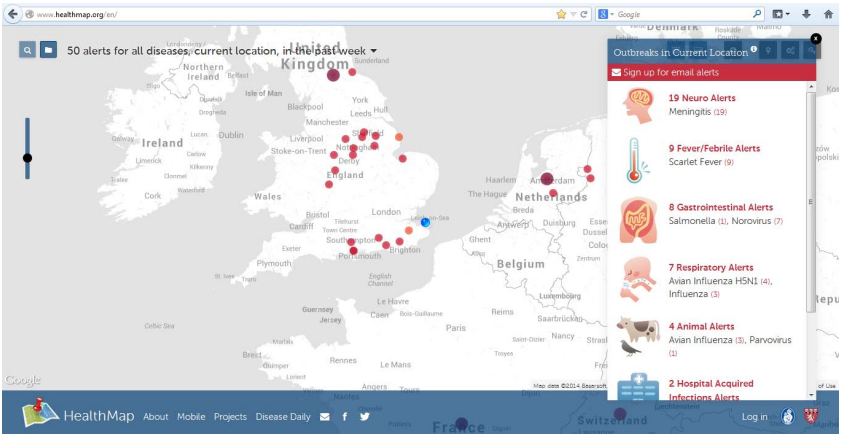


Fig. 2 HealthMap global disease alert map showing information about emerging diseases. Screenshot taken from the HealthMap Web (http://www.healthmap.org/en/), online accessed on 18th March 2014.

collects articles concerning public health in various languages from news, which are classified according to pre-defined categories as can be seen in Figure 3. Users can display world maps in which event locations are highlighted as well as statistics on the reporting about diseases, countries and combinations of them, also can apply filters for language, disease or location.

A specific and extensive application of predictive analytic techniques to public health approach are the monitoring systems of influenza through Web and social media. Google Flu Trends [6] uses Google search data to estimate flu activity during two weeks giving an early detection of disease activity, see Figure 4. This web service correlates search term frequency with influenza statistics reported by the Centers for Disease Control and Prevention (CDC), and it enables a quicker response in a potential pandemic of influenza, thus reducing its impact. Internet users perform search queries [15] and post entries in blogs using terms related to influenza illness as its diagnosis and symptoms. An increase or decrease in the number of illness searches and posts in blogs, reflects a higher or lower potential outbreak focus for influenza illness and can therefore be used to monitor it.

Finally all the systems mentioned together with their main characteristics are listed in Table 1.

3 Web Mining Solutions for Disease Outbreaks Detection

The problem of detecting and tracking epidemic outbreaks through social media can be defined as the task of extracting relevant knowledge about the epidemics in the real world given a stream of textual or multimedia data from social media. Web mining is the application of data mining techniques to discover and retrieve useful knowledge from the web documents and services. Therefore, the application of these techniques to knowledge extraction provides a better using and understanding of the data space on biomedical and health care domain [29].

There are several health data sources very useful to detect and prevent new outbreaks of different diseases. Social web media and web sites give a large amount of useful data for this purpose. Other important data sources are search engines such as Google and Yahoo! [15] [26]. In this case, the objective is to detect specific searches that involve terms that indicate influenza-like-illness (ILI) through the keywords of the queries performed. The complexity is to interpret the search context of the query, as the user may query about a particular drug, symptom or illness for a variety of reasons. Finally, ProMED-mail [28] is also a widely data source used for disease outbreaks detection. It is a human network of expert volunteers operating 24/7 as an official program of the International Society for Infectious Diseases. Their volunteers monitor global media reports and in many cases have a reporting time of outbreak disease alerts better than WHO reports.

Text mining techniques have been applied on biomedical text corpus to named entity recognition, text classification, terminology extraction, or relationship extraction [9]. These methods are a human language processing algorithms that aim to

Fig. 3 MedISys system displaying a collect of articles concerning measles in various languages. Screenshot taken from the MedISys Web (http://medusa.jrc.it/medisys/homeedition/en/home.html), online accessed on 18th March 2014.

convert unstructured textual data from large-scale collections to a specific format filtering them according to the needs.

Once the data have been extracted from the social media sites (RSS feeds, WWW, social networks, ProMED-mail, search engines, etc...), the next stage is to perform the text analysis methods for the trend detection, identifying potential sources of

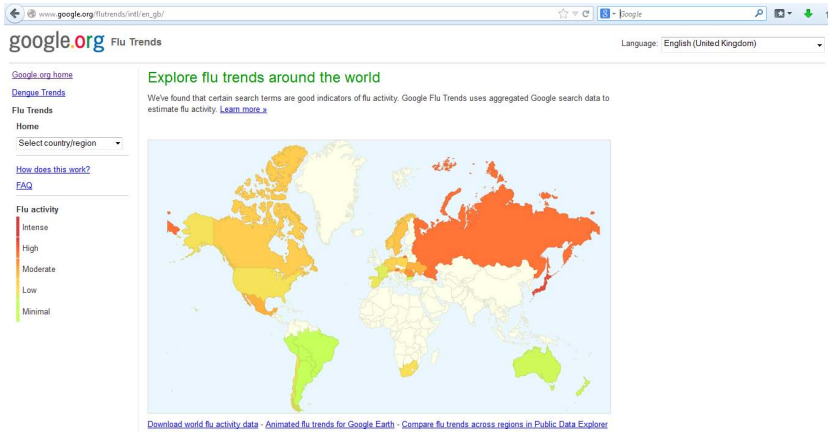


Fig. 4 Google Flu Trends to estimate flu activity during two weeks. Screenshot taken from the Google Flu Trends Web (http://www.google.org/flutrends/), online accessed on 18th March 2014.

disease outbreaks. These methods can be used to detect words related to diseases or their symptoms in published texts [20]. But this goal can be difficult because the same word can refer to a different thing depending upon context. Furthermore, a specific disease can have multiple names and symptoms associated which increases the complexity of the problem. Ontologies can help to automate human understanding of key concepts and relations between them and allow that a level of filtering accuracy can be achieved. Biomedical ontologies contain lists of terms and their human definitions, which are then given unique identifiers and classified into classes with common properties according to the specific domain treated. In the domain of EI it is necessary to identify and link term classes such as disease, symptom and species in order to detect potential focus diseases. Currently there are various available ontologies that contain all the biomedical terms necessary. For example, BioCaster ontology (BCO) [12] is in the OWL Semantic Web language to support automated reasoning across terms in 12 languages.

A new unsupervised machine learning approach to detect public health events is proposed in Fisichella et al. work [14] which can complement existing systems since it allows to identify public health events (PHE) even if no matching keywords or linguistic patterns can be found. This new approach defines a generative model for predictive event detection from document by modeling the features based on trajectory distributions.

Discovering time and location of the text is the value added by EI systems for high quality. In practice location names are often highly ambiguous because geotemporal disambiguation is so difficult, and because of the variety of ways in which cases are described across different texts. Keller et al. [19] work provides a review of the issues for epidemic surveillance and present a new method for tackling the identification of a disease outbreak location based on neural networks trained on surface feature patterns in a window around geo-entity expressions.

Finally, a different solution for outbreak detection is shown in Leskovec et al paper [22], where the problem is modelled as a network in order to detect the spreading

Table 1 Epidemic Intelligence Systems

System Name	Website	Data Sources	Description
Global Public Health Intelligence Network (GPHIN)		News wires and Web Sites	Warning tool to disease outbreaks
BioCaster	http://born.nii.ac.jp	RSS feeds, Google News, WHO, ProMED-mail and European Media Monitor	Ontology-based system for monitoring online media data
HealthMap	http://www.healthmap.org	Google News, ProMED-mail, WHO and Euro Surveillance	Global disease alert map
EpiSpider	http://www.epispider.org/	ProMed-mail and medical web sites	Human and animal disease alert map
MedISys	http://medusa.jrc.it/ medisys/homeedition/ en/home.html	Articles concerning public health from news	Monitoring tool for human and animal infectious diseases and chemical, biological, radiological and nuclear threats
Google Flu Trends	http://www.google.org/ flutrends/	Google search and CDC reports	Monitoring system of influenza

of the virus or disease as quickly as possible. They present a new methodology for selecting nodes to detect outbreaks of dynamic processes spreading over a graph. This work shows that many objective functions for detecting outbreaks in networks, such as detection time, likelihood, and population affected, are submodular. This means that, for instance, reading only a few blogs provides more new information than reading it after we have read many ones. They use this characteristic to develop an efficient approximation algorithm (CELF) which achieves near-optimal solutions and it is 700 times faster than a simple greedy algorithm.

4 Twitter Applications for Tracking and Monitoring Epidemics

The increasing popularity and use of micro-blogging services such as Twitter are recently a new valuable data source for web-based surveillance because of its message volume and frequency. Twitter users may post about an illness, and their relationships in the network can give us information about which people could be in contact with. Furthermore, user posts retrieved from the public Twitter API can come with GPS-based location tags, which can be used to detect potential disease outbreaks for a health surveillance system.

Recently, several works have already appeared shown the potential of Twitter messages to track and predict disease outbreaks. Ritterman et al. [27] work is focused on using prediction market to model public belief about the possibility that H1N1 virus will become a pandemic. In order to forecast the future prices of the prediction market, they decided to use the Support Vector Machine algorithm to carry out regression. A document classifier to identify relevant messages is presented in Culotta et al. paper [13]. In this work, Twitter messages related to flu were recollected during 10 weeks using keywords such as flu, cough, sore throat or headache. Then, several classification systems based on different regression models to correlate these messages with CDC statistics were compared, finding that the best model achieves a correlation of 0.78 (simple model regression).

Aramaki et al. [1] presents a comparative study of various machine-learning methods to classify tweets related to influenza into two categories: positive or negative. Their experimental results show that SVM model using a polynomial kernel achieves the highest accuracy (FMeasure of 0.756) and the lowest training time.

A novel real-time surveillance system to detect cancer and flu is described in paper [21]. The proposed system continuously extracts text related the two specific diseases from twitter using Twitter streaming API and applies spatial, temporal, and text mining to discover disease-related activities. The output of the three models is summarized as pie charts, time-series graphs, and US disease activity maps on the project website as can be seen in Figure 5. This system can be useful not only for early prediction of disease outbreaks, but also for monitoring distribution of different cancer types and the effectiveness of the treatments used.

Table 2 Tracking and monitoring epidemic works using Twitter data

Work Name	Machine Learning Techniques	Description
Ritterman et al.	Prediction market model and SVM	Predict flu outbreak detection
Culotta et al	Regression models	Classifier to identify flu relevant messages
Aramaki et al.	SVM using a polynomial kernel	Classifier of influenza tweets into positive or negatives
Lee et al.	Spatial, temporal, and text mining	Surveillance system to detect cancer and flu
Bodnar et al.	Regression models	Disease outbreak detection

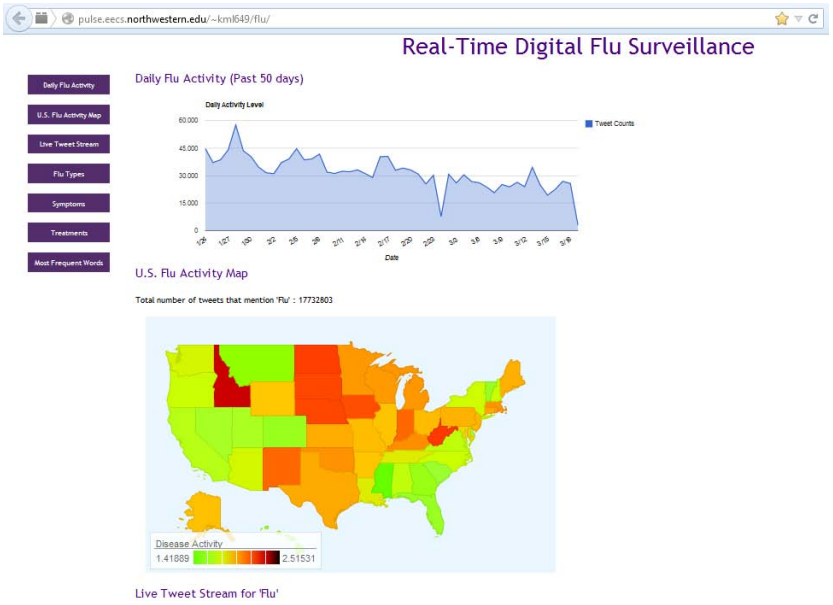


Fig. 5 A novel real-time surveillance system to detect cancer and flu from Twitter messages. Screenshot taken from the Project Web (http://pulse.eecs.northwestern.edu/ kml649/flu/), online accessed on 19th March 2014.

Well known regression models are evaluated on their ability to assess disease outbreaks from tweets in Bodnar et al. [4]. Regression methods such as Linear, Multivariable an SVM, are applied to the raw count of tweets that contain at least one of the keywords related to a specific disease, in this case "flu". The results confirmed that even using irrelevant tweets and randomly generated datasets, regression methods were able to assess disease levels comparatively well.

Finally a summary of all the systems mentioned and the machine learning techniques used is listed in Table 2. It can be noticed that most of the works use regression models, and are usually focused on detecting influenza outbreaks.

5 Discussion

All the systems and solutions presented have demonstrated the successful and beneficial use of artificial intelligence techniques when applied to extract and acquire new knowledge for public healthcare purposes. The main challenge of these systems is to interpret the search context of a particular query or document, because an user can query about a particular drug, symptom or illness for a variety of reasons. This goal can be difficult because the same word can refer to a different thing depending upon context. Furthermore, a specific disease can have multiple names and symptoms related to it, which increases the complexity of the problem. Therefore, to develop strategies for reducing false alarms and decreasing percentage of irrelevant events detected by the epidemic systems can be an important issue for future works and researches on the field.

Additionally, to identify the time and location of messages is a value added for increasing the quality of detecting possible new diseases outbreaks. But in practice location names are often highly ambiguous because geo-temporal disambiguation is so difficult, and because of the variety of ways in which cases are described across different texts.

There are several recent works show the potential of Twitter to track and detect disease outbreaks. These works demonstrate that there are health evidences in social media which can be detected. But, there can be complications regarding the possible incorrect predictions because of the huge amount of social data existing compared with the small amount of relevant data related to potential diseases outbreaks. Therefore, it is necessary to test and validate carefully all the models and methods used.

Acknowledgements. This work was supported by Spanish Ministry of Science and Education under Project Code TIN2010-19872 and Savier Project (Airbus Defence & Space, FUAM-076915)

References

1. Aramaki, E., Maskawa, S., Morita, M.: Twitter catches the flu: detecting influenza epidemics using twitter. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1568–1576. Association for Computational Linguistics (2011)
2. Asur, S., Huberman, B.A.: Predicting the future with social media. In: 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), vol. 1, pp. 492–499. IEEE (2010)
3. Bello, G., Menéndez, H., Okazaki, S., Camacho, D.: Extracting collective trends from twitter using social-based data mining. In: Bădică, C., Nguyen, N.T., Brezovan, M. (eds.) ICCCI 2013. LNCS, vol. 8083, pp. 622–630. Springer, Heidelberg (2013)
4. Bodnar, T., Salathé, M.: Validating models for disease detection using twitter. In: Proceedings of the 22nd International Conference on World Wide Web Companion, pp. 699–702. International World Wide Web Conferences Steering Committee (2013)
5. Brownstein, J.S., Freifeld, C.C., Reis, B.Y., Mandl, K.D.: Surveillance sans frontieres: Internet-based emerging infectious disease intelligence and the healthmap project. PLoS Medicine 5(7), e151 (2008)

6. Carneiro, H.A., Mylonakis, E.: Google trends: a web-based tool for real-time surveillance of disease outbreaks. Clinical Infectious Diseases 49(10), 1557–1564 (2009)
7. Chen, H., Zeng, D.: Ai for global disease surveillance. IEEE Intelligent Systems 24(6), 66–82 (2009)
8. Chen, W., Wang, C., Wang, Y.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1029–1038. ACM (2010)
9. Cohen, A.M., Hersh, W.R.: A survey of current work in biomedical text mining. Briefings in Bioinformatics 6(1), 57–71 (2005)
10. Collier, N.: Uncovering text mining: A survey of current work on web-based epidemic intelligence. Global Public Health 7(7), 731–749 (2012)
11. Collier, N., Doan, S., Kawazoe, A., Goodwin, R.M., Conway, M., Tateno, Y., Ngo, Q.H., Dien, D., Kawtrakul, A., Takeuchi, K., et al.: Biocaster: detecting public health rumors with a web-based text mining system. Bioinformatics 24(24), 2940–2941 (2008)
12. Collier, N., Goodwin, R.M., McCrae, J., Doan, S., Kawazoe, A., Conway, M., Kawtrakul, A., Takeuchi, K., Dien, D.: An ontology-driven system for detecting global health events. In: Proceedings of the 23rd International Conference on Computational Linguistics, pp. 215–222. Association for Computational Linguistics (2010)
13. Culotta, A.: Towards detecting influenza epidemics by analyzing twitter messages. In: Proceedings of the First Workshop on Social Media Analytics, pp. 115–122. ACM (2010)
14. Fisichella, M., Stewart, A., Cuzzocrea, A., Denecke, K.: Detecting health events on the social web to enable epidemic intelligence. In: Grossi, R., Sebastiani, F., Silvestri, F. (eds.) SPIRE 2011. LNCS, vol. 7024, pp. 87–103. Springer, Heidelberg (2011)
15. Ginsberg, J., Mohebbi, M.H., Patel, R.S., Brammer, L., Smolinski, M.S., Brilliant, L.: Detecting influenza epidemics using search engine query data. Nature 457(7232), 1012–1014 (2009)
16. Hartley, D.M., Nelson, N.P., Walters, R., Arthur, R., Yangarber, R., Madoff, L., Linge, J., Mawudeku, A., Collier, N., Brownstein, J.S., et al.: The landscape of international event-based biosurveillance. Emerging Health Threats 3 (2010)
17. Kamel Boulos, M.N., Sanfilippo, A.P., Corley, C.D., Wheeler, S.: Social web mining and exploitation for serious applications: Technosocial predictive analytics and related technologies for public health, environmental and national security surveillance. Computer Methods and Programs in Biomedicine 100(1), 16–23 (2010)
18. Keller, M., Blench, M., Tolentino, H., Freifeld, C.C., Mandl, K.D., Mawudeku, A., Eysenbach, G., Brownstein, J.S.: Use of unstructured event-based reports for global infectious disease surveillance. Emerging Infectious Diseases 15(5), 689 (2009)
19. Keller, M., Freifeld, C.C., Brownstein, J.S.: Automated vocabulary discovery for geo-parsing online epidemic intelligence. BMC Bioinformatics 10(1), 385 (2009)
20. Lampos, V., Cristianini, N.: Nowcasting events from the social web with statistical learning. ACM Transactions on Intelligent Systems and Technology (TIST) 3(4), 72 (2012)
21. Lee, K., Agrawal, A., Choudhary, A.: Real-time disease surveillance using twitter data: demonstration on flu and cancer. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1474–1477. ACM (2013)
22. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 420–429. ACM (2007)
23. Linge, J.P., Belyaeva, J., Steinberger, R., Gemo, M., Fuart, F., Al-Khudhairy, D., Bucci, S., Yangarber, R., van der Goot, E.: Medisys: Medical information system. In: Advanced ICTs for Disaster Management and Threat Detection: Collaborative and Distributed Frameworks, pp. 131–142 (2010)
24. Mykhalovskiy, E., Weir, L.: The global public health intelligence network and early warning outbreak detection. Canadian Journal of Public Health 97(1) (2006)

25. Paquet, C., Coulombier, D., Kaiser, R., Ciotti, M.: Epidemic intelligence: a new framework for strengthening disease surveillance in europe. Euro Surveillance: Bulletin Europeen Sur Les Maladies Transmissibles= European Communicable Disease Bulletin 11(12), 212–214 (2005)
26. Polgreen, P.M., Chen, Y., Pennock, D.M., Nelson, F.D., Weinstein, R.A.: Using internet searches for influenza surveillance. Clinical Infectious Diseases 47(11), 1443–1448 (2008)
27. Ritterman, J., Osborne, M., Klein, E.: Using prediction markets and twitter to predict a swine flu pandemic. In: 1st International Workshop on Mining Social Media (2009)
28. Victor, L.Y., Madoff, L.C.: Promed-mail: an early warning system for emerging diseases. Clinical Infectious Diseases 39(2), 227–232 (2004)
29. Xie, Y., Chen, Z., Cheng, Y., Zhang, K., Agrawal, A., Liao, W.K., Choudhary, A.: Detecting and tracking disease outbreaks by mining social media data. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, pp. 2958–2960. AAAI Press (2013)

Social Tagging Analytics for Processing Unlabeled Resources: A Case Study on Non-geotagged Photos

Tuong Tri Nguyen, Dosam Hwang, and Jason J. Jung*

Abstract. Social networking services (SNS) have been an important sources of geotagged resources. This paper proposes Naive Bayes method-based framework to predict the locations of non-geotagged resources on SNS. By computing TF-ICF weights (Term Frequency and Inverse Class Frequency) of tags, we discover meaningful associations between the tags and the classes (which refer to sets of locations of the resources). As the experimental result, we found that the proposed method has shown around 75% of accuracy, with respect to F1 measurement.

Keywords: Geotagging, Naive Bayes, Social tagging, Social networking services.

1 Introduction

Social tagging (also, called collaborative tagging) services can build a folksonomy which is a user-generated classification for resources. They have been increasingly regarded as an important research issue in social network services (SNS). There have been a number of SNS to employ the social tagging to a variety of resources (e.g., bookmarks, bibliographics, musics, and so on). Particularly, photos are the most popular resources that users want to share through SNS (e.g., Facebook, Photobucket, Instagram, Flickr and so on). In the social tagging from SNS, there have been many

Tuong Tri Nguyen · Dosam Hwang
Yeungnam University, Gyeongsan, Korea
e-mail: {tuongtringuyen,dosamhwang}@gmail.com

Jason J. Jung
Chung-Ang University, Seoul, Korea
e-mail: j2jung@gmail.com

* Corresponding author.

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_37

studies which concentrate on the two main aspects; i) to understand collective behaviors among online users, and ii) to provide online services to users [6]. Most of these studies [4, 5, 10] have commonly introduced some methods to exploit the social tagging for extracting meaningful patterns and providing various services, e.g., information searching and recommendation.

In this work, we assume that the social tagging contains spatial knowledge to differentiate the tags related to geographical locations. Thus, a geotagged folksonomy from SNS is employed i) to discover meaningful patterns between the tags and geographical locations, and ii) to estimate the location of non-geotagged resources by using the patterns.

Particularly, this study focuses on Flickr[1] (which is a well-known photo-sharing SNS) to build a geotagged folksonomy. By using either i) the tags provided from the users or ii) the geographical locations of any particular topics (e.g., names of places, persons, and events), we can extract a number of various resources (e.g., photos and videos) related to the topics [5]. Hence, for building our testing bed, we have collected the social taggings from Flickr, and put all the geographical information into the database for analyzing the information and predicting the location of non-geotagged resources.

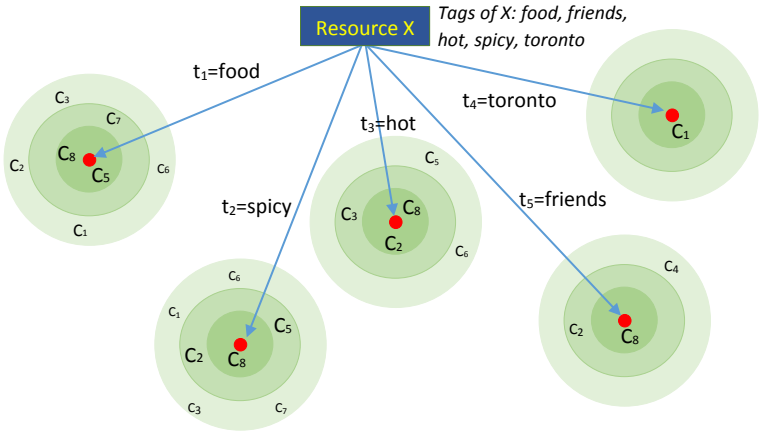


Fig. 1 Relationship between tags of resource X and a set of classes

As shown in Fig. 1, we assume that a non-geotagged resource X has 5 tags and there are 8 classes C_j (referred to as a location or a country) that X can belong to. Besides, we also know the number of occurrences of t_i in each class C_j. Thus, the question is "how can the location of X be determined?". We have to compute the weight of each tag and the probability of tags which occur in each class. In the next

[1] https://www.flickr.com

step, we determine the probability of each class and build a training set. We use the Naive Bayes method for classifying the data in the testing set.

The paper is organized as follows. Sect. 2 introduces related works. Sect. 3 shows basic knowledge and also give main steps in order to predicting location of non-geotagged resources. In the Sect. 4, experimentation has been conducted to evaluate the results of research and identify issues to be taken up for discussion. Sect. 5 draws a conclusion of this paper and indicates our future work.

2 Related Work

Several studies [1, 2] have tried to automatically collect and process the "big" data from SNS for providing the users of smart services. Some of these studies refer to tag analysis as text categorization methods [2], e.g., using tags for prediction [4] and discovering useful patterns and meaningful information from tags for recommendation. Particularly, Jung [7] exploits the tag matching to extend simple term-based queries and identify the lingual practice of each user for discovering the relationships between multilingual tags.

Also, Bischoff et al. [3] discover the associations across multiple domains and resource types and identify the gaps between the tag space and the querying vocabularies. Based on the findings of this analysis, it tries to bridge the identified gaps by focusing, in particular, on multimedia resources. By using geotagged photos, Feick and Rovertson [5] have found out a significant interaction between tag-space semantics and partial aggregation for exploring citizens sensing of urban environments (in Vancouver, Canada). Another approach [8] has used a set of geotagged photos on Flickr for extracting associative points-of-interest of a popular tourist destination in Queensland, Australia. Moreover, Clements et al. [4] is based on the Flickr geotags in the city where users have visited in order to predict a user's favorite locations in others cities and to recommend another places to the user.

Contrary to [4], we have used a set of geotagged photos with its country to determine the non-geotagged photos likely belong to the country. With this approach, we can expect to expand this research based on analyzing a set of tags of each featured country while the same refers to any problems.

3 Location Prediction by Geotagged Resources

3.1 Geotagged Folksonomy from Flickr

A folksonomy is a type of social tagging system in which the classification of data is done by users. It consists of three basic entities, which are users, tags, and resources [7]. Users create a set of tags to mark any resources, e.g., web pages,

photos, videos, and podcasts. These tags are used to manage, categorize and summarize online content. The folksonomy system also uses these tags as a way to index information, facilitate searching and navigate the resources. According to [7], a folksonomy generated by SNS is represented as $F = \langle U \times T \times R \rangle$ where R is a set of web resources described with a set of tags T by a set of users U.

Thus, as considering that some of resources are geotagged, F can be extended to the geotagged folksonomy $F^{\diamond}$.

Definition 1 (Geotagged Folksonomy). *A geotagged folksonomy is a quadruple* $F^{\diamond} = \langle U \times T \times R^{\diamond} \times \tau \rangle$, *where* $R^{\diamond} = R^{+} \cup R^{-}$ *is a set of resources. Some of them* R^{+} *are geotagged with* $\tau = \{lat, lon\}$ *which refers to the geographical coordination of the geotagged resource.*

We note that the problem of this study is to find the location τ of non-geotagged resources R^{-} by analyzing the set of geotagged resources R^{+} given from the users. For example, as shown in Fig. 1, we assume that there are 8 candidate classes (C_1 to C_8) which can potentially contain the resource $X \in R^{-}$, and we have to choose the single class as the real location of X. Thereby, the distribution of each tag of X needs to be found out.

3.2 Using TF-ICF Weight

With TF-IDF weight, we obtained the results according to what about discussion above, i.e., they will return class that has the highest probability (e.g., class 8). But, with ICF weight value, we achieved more accurate classifier. The value classified will not be class 8 such as a result of the TF-IDF. The classification by ICF weight returns class 1, and this is correct class. We can see the illustration in Fig. 2. Although only 3 tags belonging to class 1, but it is being correctly classified by TF-ICF.

Using TF-IDF to compute the term weight based on two statistics, term frequency (TF) and inverse document frequency (IDF) which are very popular in fields such

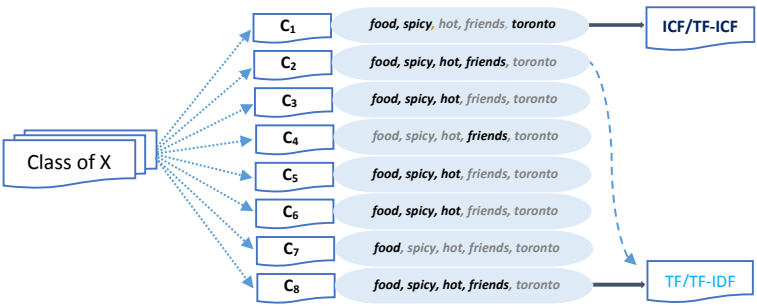


Fig. 2 Predicting location from a set of tags of the resource by TF-ICF

as classifying document [12]. They could determine the exact values of class of the document by using term frequency from set of words in that document.

For this purpose, we use geotagged resources to reach an new method for classifying the data resources from social network system. We use the location of the geotagged resources to determine the locations of the non-geotagged ones. We consider to use each location as a class for classifying. Thus, the classification problem is specified on determining location (or country) of non-geotagged photos on Flickr.

We denoted the terms as follows:

- $P(C)$ is a set of photos $\{X_1, X_2, .., X_n\}$;
- $P(C_i)$ is a set of photos of C_i;
- $P(t)$ is a set of photos tagged by t;
- $P(t, C_i)$ is a set of photos of C_i tagged by t;
- C is a set of classes (which are referred to as a set of countries);
- C_i is a class i-th, $i \in [1, m]$;
- m is the total number of classes;
- T is a set of tags;
- $T(C_i)$ is a set of tags of C_i;
- X is a photo with n tags, $X = \{t_1, t_2, .., t_n\}$;
- $\wp$ is a probability function

While the value of TF weight $\in [0, 1]$ in [11,12], this work uses TF weight $\in [0.5, 1]$ (since it is convenient for combining with Naive Bayes method). TF can be is computed by

$$tf(t_i, C_j) = 0.5 + 0.5 \times (\frac{P(t_i, C_j)}{max\{P(t_k, C_j)|t_k \in T_{C_j}\}}) \quad (1)$$

and, IDF is determined as

$$idf(t_i, C) = 1 + log(\frac{|C|}{1 + |\{C_j \in C|t_i \in T_{C_j}\}|}) \quad (2)$$

where $|C|$ is cardinality of C, or total of class in training set, $|\{C_j \in C : t_i \in T_{C_j}\}|$ is number of class contains tag t_i. TF-IDF can be computed as

$$tfidf(t_i, C_j) = tf(t_i, C_j) \times idf(t_i, C) \quad (3)$$

While we use IDF for reducing the value of tags occurred in many classes, we use formula ICF for determining exactly class frequency with tag. From IDF and coefficient, ICF can be represented as

$$icf(t_i, C_j) = (1 + log(\frac{P(t_i, C_j) + 1}{(|\{C_k \in C : t_i \in T_{C_k}\}| + 1)})) \times idf(t_i, C) \quad (4)$$

The ICF value of tags which only occur in one class (e.g., the tag 'Toronto' as shown in Fig. 1 get high ICF value as shown in Tab. 4) became useful for classifying and predicting process. Finally, TF-ICF can be computed by

$$tficf(t_i, C_j) = tf(t_i, C_j) \times icf(t_i, C_j) \tag{5}$$

3.3 Using Naive Bayes Method for Classification Problem

According to Naive Bayes theorem [11], we compute the probability of resource X is contained by class C_i by

$$\wp(C_i|X) = \frac{\wp(X|C_i)\wp(C_i)}{\wp(X)} \tag{6}$$

where $\wp(C_i|X)$ is probability of class i, contains resource X, $\wp(C_i)$ is probability of class i, $\wp(t_k|C_i)$ is probability of tag t_k in class i, $\wp(X|C_i)$ is probability of resource X in class i, and $\wp(t_k)$ is probability of tag t_k.

Here, we only consider $X = \{t_1, t_2, .., t_n\}$ and each t_j is independent with each other. Thus, $\wp(X|C_i) = \sum_{j=1}^{n} \wp(t_j|C_i)$ where n is the number of tags of resource X.

Using Equ. 6, we compute the class of resource X by getting the max value of $\wp(C_i|X)$, with $i \in [1, m]$, where m is the number of classes in training dataset. Besides, since probability value of each class has to be divided by the same value ($\wp(X)$), we can omit the denominator. We show the value of classifying probability of resource X as

$$classOf(X) = \arg\max\{\wp(X|C_i)\wp(C_i)\}. \tag{7}$$

3.4 Proposed Algorithm

To propose the classifying algorithm, we have considered using TF-ICF weight of tags by using the Naive Bayes-based classification method. By comparing to the similar work [3, 11], we propose a novel classification algorithm as follows:

In this algorithm, the training set is used in order to compute the tag weight (TF, ICF, TF-ICF and TF-IDF) for the Input set. The testing set is used to predict the location of resource and to evaluate the results. The algorithm used the Naive Bayes method to compute the probability of each class.

Algorithm 6. Algorithms for classification

Data: Training set, Testing set
Result: Geotags for Testing set
initialization;
Compute $\wp(C_i)$;
$\wp(C_i|X) = 0$;
while *photo* $X \in P_{Testing}$ **do**
  while *tag* $t_j \in T_X$, *and* $t_j \in T_{Training}$ **do**
    if $t_j \in T(C_i)$ **then**
      $\wp(C_i|X) = \wp(C_i|X) + w(t_j) \times \wp(t_j|C_i)$
    else
      $\wp(C_i|X) = \wp(C_i|X) + w(t_j) \times (1 - \wp(t_j|C_i))$
    end
  end
  Class.of(X) $\Leftarrow$ arg-max$\{\wp(C_i)\wp(C_i|X))\}$
end
{with $w(t_j)$ is TF, ICF or TF-ICF value of tag t_j;}

4 Experimentation

4.1 Dataset

We collected data from Flickr, and performed some basic data processing to obtain the data, as the basis for the experiments. As shown in Tab. 1, 4 keywords were selected to collect the dataset. On average, more than 12 tags per each photo and less than 20% of the collected photos have been geotagged.

Moreover, we also used the threshold for removing geotagged photos, if they can not create a new class (we assume that each class has more than 10 photos).

Table 1 Collecting dataset

Keyword	# photos on Flickr	# photos collected	# Geotagged photos	# Tags
kimchi	16,143	8,490	1,136	100,144
noodle	49,128	10,779	1,527	143,766
samsung	442,372	1,142	254	13,808
tower	1,413,010	6,499	2,528	114,768

After collecting data, we split them into two sets (70% in training set, 30% in testing set). We analyze the data in training set. As an example with keyword 'kimchi', we could determine 8 classes in Tab. 2.

We implemented a simple preprocessing in order to remove some tags which have no meaning (e.g., stop words) [7] and counted the number of tags for each class (some popular tags are showed as in Tab. 3).

Table 2 Extracting data with 'kimchi'

Class	# Photos	# Tags	Class	# Photos	# Tags
Canada (CA)	12	90	South Korea (KR)	248	2917
China (CN)	18	300	Taiwan (TW)	11	162
Japan (JP)	22	271	United Kingdom (UK)	52	426
North Korea (NK)	149	1520	United States (US)	283	3552

Table 3 Popular tags with 'kimchi'

Tag/Class	Canada	China	Japan	NorthKorea	SouthKorea	Taiwan	UnitedKingdom	UnitedStates
korean	6	9	9	0	96	1	37	178
hot	0	7	5	0	1	1	0	7
spicy	2	5	2	0	7	1	0	13
food	7	9	12	0	138	5	47	131
friends	0	9	0	2	0	0	0	32

4.2 Experimental Results

We compute the the value of all tags which include probability, TF, ICF, TF-IDF, TF-ICF and put them into the dataset as shown in Tab. 4.

On the following step, we use the Alg. 6 to implement. The classified results are computed the precision, recall values. Here, we use equation in [9] to calculate the F-measure values.

We implemente on the dataset with 10 iterations for the input data 10%, 20%, .. to 100% (of training set). For each iteration, we use testing set to predict location and to compute the results. The process are conducted following 3 steps (illustrations by computing value of resource X on the Fig. 1).

1. Computing $\wp(C_i)$, $\wp(t_k)$ and $\wp(t_k|C_i)$: We compute the value for classifying with TF/ICF/TF-ICF weight for resource X, given by Tab. 4 as follows:

Table 4 The results for classification

Class	food		friends		hot		spicy		toronto		Results		
	TF	ICF	TF	ICF	TF	ICF	TF	ICF	TF	ICF	TF	ICF	TF-ICF
CA	0.017	0.044	0.010	0.036	0.019	0.049	0.048	0.147	0.750	9.085	0.816	9.278	6.935
CN	0.021	0.059	0.163	0.979	0.213	0.856	0.106	0.362	0.071	0.340	0.504	2.257	1.605
JP	0.028	0.081	0.010	0.036	0.141	0.594	0.045	0.147	0.071	0.340	0.215	0.823	0.50
NK	0.001	0.002	0.033	0.197	0.019	0.049	0.013	0.031	0.071	0.340	0.033	0.197	0.101
KR	0.303	1.316	0.010	0.036	0.038	0.137	0.114	0.521	0.071	0.340	0.456	1.975	1.361
TW	0.012	0.031	0.010	0.036	0.041	0.137	0.030	0.085	0.071	0.340	0.084	0.254	0.144
UK	0.128	0.385	0.010	0.036	0.019	0.049	0.013	0.031	0.071	0.340	0.128	0.385	0.367
US	0.271	1.241	0.399	4.278	0.157	0.856	0.203	1.038	0.071	0.340	1.031	7.414	4.270

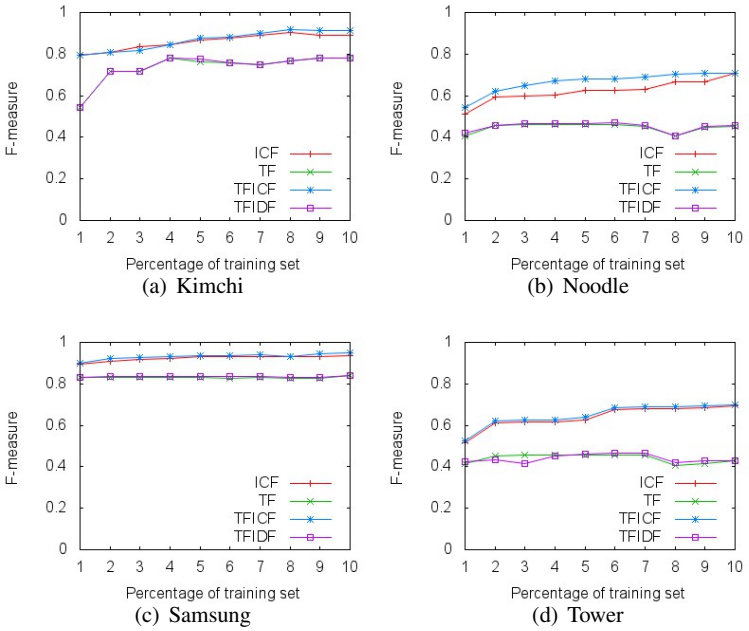


Fig. 7 Compare $F_{measure}$ with some keywords

2. Classifying for new resource $X = (t_1, t_2, \ldots, t_n)$: We calculate the probability of each class to know the country of resource X. The classified value of X is computed by using the Equ. 7 and the results are showed in Tab. 4.
3. Computing $F_{measure}$: It is used on [9], $F_{measure} = \frac{2PR}{P+R}$, where P is the precision and R is the recall.

We show the implementation results in the Fig. 7 and Fig. 8.

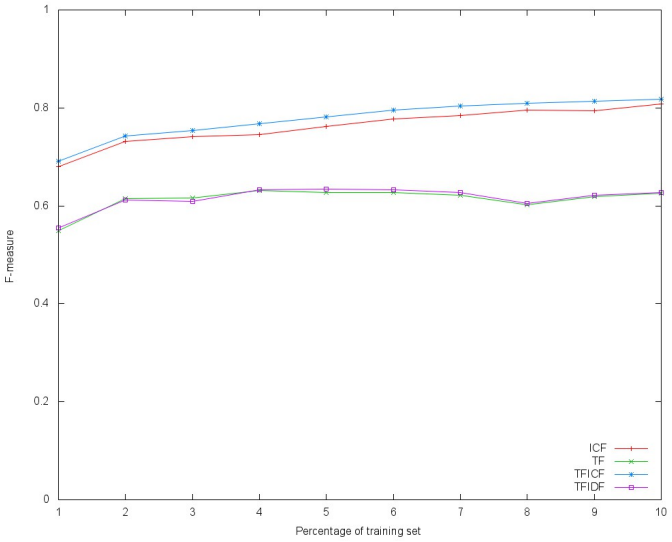


Fig. 8 Average of $F_{measure}$ with all keywords

5 Conclusion and Future Work

Predicting location of resources based on its tags might be used for classifying, categories or clustering with each place, country or city. In this paper, we implement with different keywords such as “kimchi”, “noodle”, “samsung” and “tower”. We got the results approximate more than 0.75 with F-measure value. However, we also known that the results have depended on the data of each searching keyword and also depended on the variability of the tags in training set.

Through this work, we have found out there are many issues that need further consideration, such as the construction process of the training data set, collecting data should be using multi-lingual search. Besides, the issue of handling the selected tags for the classifying should be also considered.

We improve our research by expanding the set parameters such as user data and some others collected attributes. On the other hand, we will use collected data by searching multi-language keyword same as the method which is used in [6]. We are planning

1. to combine more folksonomies which are available on the web (e.g user, owner),
2. to consider proposing new approach to recommend on SNS based on our results and
3. to rank the location through set of tags.

Acknowledgements. This work was supported under the framework of international cooperation program managed by National Research Foundation of Korea (NRF-2013K2A1A205-5213). Also, this work is supported by BK21+ of National Research Foundation of Korea.

References

1. Atzori, L., Iera, A., Morabito, G.: The internet of things: A survey. Computer Networks 54(15), 2787–2805 (2010)
2. Atzori, L., Iera, A., Morabito, G., Nitti, M.: The social internet of things (siot) when social networks meet the internet of things concept, architecture and network characterization. Computer Networks 56(16), 3594–3608 (2012)
3. Bischoff, K., Firan, C.S., Nejdl, W., Paiu, R.: Bridging the gap between tagging and querying vocabularies: Analyses and applications for enhancing multimedia {IR}. Web Semantics: Science, Services and Agents on the World Wide Web 8(2-3), 97–109 (2010)
4. Clements, M., Serdyukov, P., de Vries, A.P., Reinders, M.J.: Using flickr geotags to predict user travel behaviour. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, pp. 851–852. ACM (2010)
5. Feick, R., Robertson, C.: A multi-scale approach to exploring urban places in geotagged photographs. Computers, Environment and Urban Systems (2014)
6. Jung, J.J.: Discovering community of lingual practice for matching multilingual tags from folksonomies. The Computer Journal 55(3), 337–346 (2012)
7. Jung, J.J.: Cross-lingual query expansion in multilingual folksonomies: A case study on flickr. Knowledge-Based Systems 42(0), 60–67 (2013)
8. Lee, I., Cai, G., Lee, K.: Exploration of geo-tagged photos through data mining approaches. Expert Systems with Applications 41(2), 397–405 (2014)
9. Manning, C.D.: Foundations of statistical natural language processing, vol. 999. MIT Press
10. Morrison, P.: Tagging and searching: Search retrieval effectiveness of folksonomies on the world wide web. Information Processing and Management 44(4), 1562–1579 (2008)
11. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys (CSUR) 34(1), 1–47 (2002)
12. Zhang, W., Yoshida, T., Tang, X.: Tfidf, lsi and multi-word in information retrieval and text categorization. In: IEEE International Conference on Systems, Man and Cybernetics, SMC 2008, pp. 108–113. IEEE (2008)

2D-Social Networks: A Way to Virally Distribute Popular Information Avoiding Spam

Pasquale De Meo, Fabrizio Messina, Domenico Rosaci, and Giuseppe M.L. Sarné

Abstract. In Online Social Networks, in order to virally distribute some topics and, at the same time, protecting users from undesired messages, we propose to diffuse viral campaigns only on a second dimension of the social network. In the proposed approach, software agents assist the user by selecting the most appropriate campaigns for their owners. A users-to-campaigns matching algorithm, called **V**iral **F**iltered **D**iffusion, allows the agents to dynamically manage the evolution of the viral activity. Preliminary experiments clearly show the advantages in assigning to the users only campaigns compatible with their orientations.

1 Introduction

The viral diffusion of contents on online Social networks (SNs) is largely studied for its social, political and economic implications [1,23,27]. However, the opportunity to distribute contents on OSNs can represent an effective tool for social spam campaigns or for delivering malware contents [4,9,10], since a user responds more readily to a message coming from a friend. In other words, the viral diffusion of certain topics is a desirable for their producers but undesirable for the users not interested to them.

Fabrizio Messina
DMI, Catania University, V.le A. Doria 6, 95125 Catania, Italy
e-mail: messina@dmi.unict.it

Pasquale De Meo
DICAM, University of Messina, 98166 Messina, Italy
e-mail: pdemeo@unime.it

Domenico Rosaci · Giuseppe M.L. Sarné
{DIIES, DICEAM}, University "Mediterranea" of Reggio Calabria,
Loc. Feo di Vito, 89122 Reggio Calabria, Italy
e-mail: {domenico.rosaci,sarne}@unirc.it

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_38

Currently, several researches to detect spam in OSNs have been presented [9, 12, 22, 29] but, at the best of our knowledge, any of them allows viral topics diffusion also protecting users from spam campaigns. To this aim, we propose to build upon the real SN of users a *twin* SN of agents. In such a 2D-Social Network, agents automatically manage the viral topics distribution avoiding the spam diffusion, while the real SN users can access to the viral content on the twin SN. A distributed algorithm called **V**iral **F**iltered **D**iffusion (VFD) exploiting a dissimilarity measure and performed by the agents of the twin OSNs, (*i*) matches the content of the viral messages with the users' profiles, (*ii*) filtering undesired messages for the single user u and (*iii*) automatically distributing the messages of interest for u to his friends.

Preliminary experiments performed on a simulated 2D-Social Network clearly show the improvements introduced by our approach to the users' satisfaction being reasonably assigned to the users only campaigns compatible with their interests.

The paper is organized as follows. Section 3 introduces the proposed scenario, the multi-agent architecture and describes the VFD Algorithm. In Section 3 we discuss related literature, while Section 4 presents the experiments performed to validate our proposal and some conclusions.

2 The 2D-Social Network Scenario and the VFD Algorithm

We define a *Social Network* S as a pair $\langle U_S, f_S \rangle$, where U_S is a set of users and f_S is a mapping taking a user $u \in U_S$ as input and returning a set of users to represent the *friends* of u as output. Moreover, let a *2D Social Network* be a pair $S^{2D} = \langle S, S^* \rangle$, where S and S^* are two OSNs and for each user $u \in U_S$ also $u \in U_{S^*}$. Note that a user v can be a friend of u in S but not in S^*, and vice-versa.

We assume that each SN user u is associated with an agent a_u which interacts with other agents by means of a *Communication Layer* and a naming service associated to the whole SN, i.e. the *Directory Facilitator* agent (DF). In the above context, each user u of S^{2D} is characterized by the properties:

- Let CI be the set of all the categories of interests of u in S and S^* (e.g. *music, sport, etc*), and each element $\gamma \in CI$ is a string denoting a category. Let I_u be a mapping $\forall \gamma \in CI$ returning $I_u(\gamma)$, ranging in $[0..1] \in \mathbb{R}$, to represent the u's level of interest on the category of interests based on her/his actual behaviour.
- Let a behaviour be a type of action that u could performs in S and S^*, as publishing more than 3 posts/hour or posts longer than 400 chars. We represent each behaviour by a set $B = \{b_1, b_2, \ldots, b_n\}$ and the overall u's behaviours with respect to B by a set of boolean variables $B_u = \{b_{u,1}, b_{u,2}, \ldots, b_{u,n}\}$.

Moreover, a profile $p_u := \langle I_u, B_u \rangle$ is associated to each user u, and the agent a_u automatically updates the profile p_u of its own user u as follows:

- For each u's action linked to a category γ, then a_u updates $I_u(\gamma)$, as $I_u(\gamma) = \alpha \cdot I_u(\gamma) + (1 - \alpha) \cdot \delta$ based on its past value and a share for the u's action,

where $\alpha, \delta \in [0,1] \in \mathbb{R}$. Specifically, α is the relevance that u gives to the past $I_u(\gamma)$ values with respect to the increment δ that u gives to his/her interest in c for her/his action. The $I_u(\gamma)$ updating criteria is effectively used in many multi-agent approaches when the weights (e.g. α, δ) are correctly set [5, 25, 26].

- Each way the user u performs an action on S, its agent a_u analyses the action and sets the appropriate boolean values for all the variables contained in B_u. Analogously, if the user u adds a new friend in her/his friend list, or delete an existing friend from this friend list, then the agent a_u consequently updates f_u.

When a user starts with a campaign c, a new *viral* agent a_c is stored into the DF. We define a property B_c similar to B_u to profile the overall behaviour of the users involved in c and the agent a_c manages a profile $p_c := \langle I_c, B_c \rangle$ to: (*i*) Ask the values $I_c(\gamma)$, for each category $\gamma \in CI$ to the agent a_u of the user u that created the campaign, where $I_c(\gamma) \in [0,1] \in \mathbb{R}$, represents the "weight" of γ_c; (*ii*) Analyse, informed by a_u, each u's action related to the campaign c to set the appropriate boolean values contained in B_c (the elements B_c have the same mean that for B_u).

We discuss the VFD (Viral Filtered Diffusion) as a set of activities executed by user agents a_u and campaign agents a_c. Let C be the set of the n viral campaigns where u is engaged, with $n \leq n_{max}$, being n_{max} the maximum number of campaigns for a user. Let T be the (constant) period between two consecutive execution of a task, called *epoch*, by an agent and let m be the number of the viral agents contacted by a_u at each epoch. We suppose that a_u records into an its cache the profile p_c of the campaign $c \in C$ and the date of acquisition (d_c). As a consequence, each user agent a_u periodically executes the *viral diffusion tasks* and each time it receives a viral message, it performs the *filtering message tasks* as described below.

The Viral Diffusion Tasks. Periodically a_u executes the following set of tasks.

- Randomly selects in the DF repository m viral agents (a_c) associated with campaigns not present in C. Let Y be the set of the campaigns associated with these viral agents, and let $Z = C \bigcup Y$ a set of all the viral agents present in C or in Y.
- For each campaign $c \in Y$, (i.e. for each campaign $c \in C$) such that the date of acquisition d_c is higher than a fixed threshold date ψ, it sends a message to the viral agent a_c, for requesting the profile p_c managed by a_c for the campaign c.
- computes a *dissimilarity measure* between the profiles of u and of the campaign c for each received p_c, defined as a weighted mean of two contributions c_I and c_B, associated with the properties I and B, measuring how much are different their values. To this aim: (*i*) c_I is computed, on all the categories present in the SN, as $c_I = (\sum_{\gamma \in CI} |I_u(\gamma) - I_c(\gamma)|)/|C|$; (*ii*) c_B is the average on all the values 0/1 associated with each pair of equal/different boolean variables in B_u and B_c; (*iii*) The dissimilarity d_{uc} of a campaign c with respect to u is then computed as $d_{uc} = (w_I \cdot c_I + w_B \cdot c_B)/(w_I + w_B)$, where $w_I, w_B \in [0,1] \in \mathbb{R}$.
- Now, let $\tau \in [0,1] \in \mathbb{R}$ be a threshold such that a viral campaign c is eligible if $d_{uc} > \tau$. Then, a_u inserts in a set $PASS$ those campaigns $c \in Z : d_{uc} > \tau$ (if more than n_{max} campaigns satisfy this condition, then the n_{max} campaigns

having the highest values of global difference are selected). For each campaign $c \in PASS$ and for each friend $v \in S^*.f(u)$ not contacted on c, then a_u sends a viral message about c to a_v. The agent a_u deletes each campaign $c \in C : c \notin PASS$.

The Filtering Message (FM) Tasks. The following set of tasks is periodically executed by a_u at each time it receives a viral message concerning a campaign c.

- If $c \notin C$ and at least a campaign $l \in C : d_{uc} < d_{ul}$ exists, then c is added to C and the campaign $z \in C$ having the smallest dissimilarity is deleted from C.
- For each friend $v \in S^*.f(u)$ not contacted on c, the agent a_u sends a viral message to the agent a_v having as subject the campaign c.

3 Related Work

The personalization of the SN services provided to the members [8], is considered a success key. To this aim, SNs need to know interests and preferences of the users [1, 7], often by exploiting software agents managing a user profile updated based on his/her activities, as in our proposal, and used for example, to provide suggestions about media contents [28] or groups to join with [24], advertising goods [21], improving groups homogeneity [15–17] or spam detection [12, 22, 29].

Storing data about billions of users, SNs are the best medium for viral campaigns being users orientations easily exploiting to plan viral campaigns on the best target users. Many researches investigated on selecting the most efficient set of nodes used to diffuse viral SN campaigns by studying diffusion models [6, 13, 14], nodes selection algorithms and viral strategies [2, 3, 11]. As negative effects viral campaigns can spread spam or malware [9] and spam models and anti-spam strategies have been evaluated in [12, 29] for instance in [22] is proposed an anti-spam algorithm based on the use of white and black list. Compared to the works discussed above, our approach is able to allow viral topics diffusion but avoiding spam campaigns.

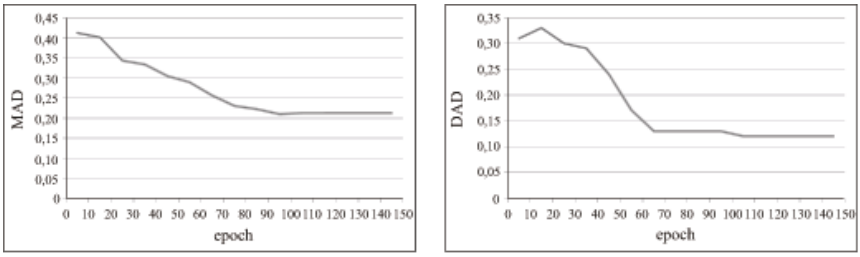


Fig. 1 The variation of MAD and DAD vs epochs

4 Experiments and Conclusions

Some experiments have been performed by using the JADE platform, to evaluate the effectiveness of the VFD algorithm by considering a OSN having 120.000 users and 200 viral campaigns. Each user's profile has been generated as follows: (*i*) each values $I_u(c)$ is a random value from a uniform distribution on $[0..1]$; (*ii*) B_u contains 5 boolean random values, representing 1-the user's attitude to publish less than 4 post per day, 2-posts longer than 150 chars, 3-comments refered to the posts of the other users at least three times per day, 4-answer to the comments refered to its own posts and 5-messages of other users.

Initially, 10 viral campaigns are assigned to each user, their profiles are assigned by randomly generating their I and B and the parameters introduced in Section 3 are set to $\alpha = 0.5$, $\delta = 0.1$, $\psi = 10$, $\tau = 0.7$. To evaluate our algorithm, we use the *average dissimilarity* between each pair of objects in a cluster (in our scenario equivalent of a cluster of the campaigns in which u is engaged, denoted as C_u). The average dissimilarity of u (AD_u) is computed as $AD_u = (\sum_{c \in C_u} d_{uc})/(|C_u|)$. The global average dissimilarity of the users is computed by mean of the couple mean $MAD = \frac{\sum_{u \in U} AD_u}{|G|}$ and standard deviation $DAD = \sqrt{\frac{\sum_{u \in U}(AD_u - MAD)^2}{|U|}}$ of all the AD_c. Starting from a generation of inhomogeneous population of users and campaigns profiles ($MAD = 0.412$, $DAD = 0.31$) we applied the VFD algorithm for 150 epochs (each one simulating a day) for each user. The simulation results are reported in Figure 1. We observe that the VFD algorithm achieves a stable configuration after 135 epochs ($MAD = 0.213$ and $DAD = 0.12$), which means that a relevant decrement in time, in terms of MAD and DAD, is introduced by the VFD algorithm.

Preliminary experiments presented above clearly support our idea to spread viral campaigns only on a second dimension of the OSN. The assistance provided by the software agents allows the users to select only the most appropriate campaigns, and to forward them to their own friends. The execution of the **V**iral **F**iltered **D**iffusion (VFD) significantly decremented the heterogeneity between users and campaigns. In the future, we are planning to perform a set of targeted simulations on real SN data by using a simulator [18–20] able to simulate even billions of nodes.

Acknowledgements. This work is a part of the research project **PRISMA**, code **PON04a2_A/F**, funded by the Italian Ministry of University within the **PON 2007-2013** framework program.

References

1. Ardon, S., et al.: Spatio-temporal and events based analysis of topic popularity in twitter. In: CIKM, pp. 219–228. ACM (2013)
2. Barbieri, N., Bonchi, F., Manco, G.: Cascade-based community detection. In: Proceedings of the Sixth ACM Int. Conf. on Web Search and Data Mining, pp. 33–42. ACM (2013)

3. Belák, V., Lam, S., Hayes, C.: Towards maximising cross-community information diffusion. In: Advances in Social Networks Analysis and Mining, 2012, pp. 171–178. IEEE (2012)
4. Benevenuto, F., Magno, G., Rodrigues, T., Almeida, V.: Detecting spammers on twitter. In: Collaboration, Electronic Messaging, Anti-Abuse and Spam Conf., vol. 6, p. 12 (2010)
5. Buccafurri, F., Palopoli, L., Rosaci, D., Sarné, G.M.L.: Modeling cooperation in multi-agent communities. Cognitive Systems Research 5(3), 171–190 (2004)
6. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: Proc. of the 15th Int. Conf. on Knowledge Discovery and Data Mining, pp. 199–208. ACM (2009)
7. De Meo, P., Ferrara, E., Abel, F., Aroyo, L., Houben, G.J.: Analyzing user behavior across social sharing environments. ACM Trans. on Intelligent Systems and Technology (TIST) 5(1), 14 (2013)
8. De Meo, P., Nocera, A., Rosaci, D., Ursino, D.: Recommendation of reliable users, social networks and high-quality resources in a social internetworking system. Ai Communications 24(1), 31–50 (2011)
9. Gao, H., Hu, J., Wilson, C., Li, Z., Chen, Y., Zhao, B.Y.: Detecting and characterizing social spam campaigns. In: Proc. 10th Conf. on Internet Measurement, pp. 35–47. ACM (2010)
10. Ghosh, S., Viswanath, B., Kooti, F., Sharma, N.K., Korlam, G., Benevenuto, F., Ganguly, N., Gummadi, K.P.: Understanding and combating link farming in the twitter social network. In: Proc. of the 21st Int. Conf. on World Wide Web, pp. 61–70. ACM (2012)
11. Goyal, A., et al.: On minimizing budget and time in influence propagation over social networks. Social Network Analysis and Mining 3(2), 179–192 (2013)
12. Heymann, P., Koutrika, G., Garcia-Molina, H.R.: Fighting spam on social web sites: A survey of approaches and future challenges. IEEE Internet Computing 11(6), 36–45 (2007)
13. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing spread of influence in a social network. In: Proc. 9th Int. Conf. on Knowledge Discovery and Data Mining, pp. 137–146. ACM (2003)
14. Kempe, D., Kleinberg, J.M., Tardos, É.: Influential nodes in a diffusion model for social networks. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 1127–1138. Springer, Heidelberg (2005)
15. Messina, F., Pappalardo, G., Rosaci, D., Santoro, C., Sarné, G.M.L.: HySoN: A distributed agent-based protocol for group formation in online social networks. In: Klusch, M., Thimm, M., Paprzycki, M. (eds.) MATES 2013. LNCS, vol. 8076, pp. 320–333. Springer, Heidelberg (2013)
16. Messina, F., Pappalardo, G., Rosaci, D., Santoro, C., Sarné, G.M.L.: A Trust-Based Approach for a Competitive Cloud/Grid Computing Scenario. In: Fortino, G., Badica, C., Malgeri, M., Unland, R. (eds.) IDC 2012. SCI, vol. 446, pp. 129–138. Springer, Heidelberg (2012)
17. Messina, F., Pappalardo, G., Rosaci, D., Santoro, C., Sarné, G.M.L.: A Distributed Agent-Based Approach for Supporting Group Formation in P2P e-Learning. In: Baldoni, M., Baroglio, C., Boella, G., Micalizio, R. (eds.) AI*IA 2013. LNCS, vol. 8249, pp. 312–323. Springer, Heidelberg (2013)
18. Messina, F., Pappalardo, G., Santoro, C.: Complexsim: An smp-aware complex network simulation framework. In: 2012 Sixth International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), pp. 861–866. IEEE (2012), doi:10.1109/CISIS.2012.102
19. Messina, F., Pappalardo, G., Santoro, C.: Exploiting gpus to simulate complex systems. In: 2013 Seventh International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS), pp. 535–540. IEEE (2013), doi:10.1109/CISIS.2013.97
20. Messina, F., Pappalardo, G., Santoro, C.: Complexsim: a flexible simulation platform for complex systems. International Journal of Simulation and Process Modelling 8(4), 202–211 (2013), doi:10.1504/IJSPM.2013.059417
21. Nascimento, V., et al.: Exploring emergent social networks to improve agent mediated e-commerce. In: 10th Int. Conf. on e-Business Engineering, pp. 50–55. IEEE (2013)
22. Oscar, P., Roychowdbury, V.P.: Leveraging social networks to fight spam. IEEE Computer 38(4), 61–68 (2005)
23. Rajyalakshmi, S., Bagchi, A., Das, S., Tripathy, R.M.: Topic diffusion and emergence of virality in social networks. CoRR, abs/1202.2215 (2012)

24. Rosaci, D., Sarné, G.M.L.: Matching Users with Groups in Social Networks. In: Zavoral, F., Jung, J.J., Badica, C. (eds.) IDC 2013. SCI, vol. 511, pp. 45–54. Springer, Heidelberg (2013)
25. Rosaci, D., Sarnè, G.M.L.: Multi-agent technology and ontologies to support personalization in B2C e-Commerce. Electronic Commerce Research and Applications 13(1), 13–23 (2014)
26. Rosaci, D., Sarnè, G.M.L., Garruzzo, S.: Integrating trust measures in multiagent systems. International Journal of Intelligent Systems 27(1), 1–15 (2012)
27. Ruhela, A., et al.: Towards the use of online social networks for efficient internet content distribution. In: ANTS, pp. 1–6. IEEE (December 2011)
28. Shin, S., et al.: The user-group based recommendation for the diverse multimedia contents in the social network environments. In: 9th Int. Conf. on DASC, pp. 202–206. IEEE (2011)
29. Stringhini, G., Kruegel, C., Vigna, G.: Detecting spammers on social networks. In: Proc. of the 26th Annual Computer Security Applications Conf., pp. 1–9. ACM (2010)

The Effect of Topology on the Attachment Process in Trust Networks

V. Carchiolo, A. Longheu, M. Malgeri, and G. Mangioni

Abstract. The massive use of web based social networks endorse trustworthiness to establish reliable relationships. Trust is often used to rank entities hence during the attachment process in a trust network, newcomers aim at improving their rank. Moreover, since in real systems each new link implies some cost for the node, we also consider how many links (i.e. how much effort) a node must exert in order to achieve a given rank. In this work, the rank-effort relationship in networks with a high number of nodes and different topologies - random and scale-free - also in presence of communities is considered, in order to examine how the topology affects the attachment process. Results show that the behavior is similar with differences on the effort required to get the same rank in different topologies; in addition, a good rank can be achieved with a considerably less effort than the best rank, thus a satisfactory rank-effort tradeoff can be found for each topology.

1 Introduction

The massive use of web based social networks [1] and the big amount of personal data they allow to share leverage trust as the underlying mechanism to establish *reliable* relationships. Trust has a long story and different meanings depending on the discipline being considered, as sociology, psychology, economics and politics. Significant contributions in computer science can be found [2, 3], together with recent applications within social networks [4], viral marketing, collaborative filtering, security, business analytics and many others (e.g. [5]).

A common assumption is that trust allows to rank entities each time a choice is required. Hence, whenever a new node joins a trust network it wishes to establish links with others but at the same time it chooses those links that improve its rank.

V. Carchiolo · A. Longheu · M. Malgeri · G. Mangioni
Dipartimento di Ingegneria Elettrica Elettronica e Informatica,
Università degli Studi di Catania, Italy
e-mail: {alessandro.longheu,vincenza.carchiolo,michele.malgeri,
giuseppe.mangioni}@dieei.unict.it

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_39

Moreover, nodes ranking is also affected by networks changes, for instance just those due to the new link(s) the node establishes, therefore it is also important how to preserve or even strengthen trust as the network evolves.

Given this premises, in this paper we consider the attachment process in a trust network. Specifically, we suppose the new node initially links to the one that provides it with the highest possible rank; at each step, it adds a new link with the same criterion, trying to increase (or at least to preserve) its rank. This heuristic aims at limiting the number of new links as much as possibile, indeed in real systems each link implies some cost, e.g. in P2P networks peers must collect positive downloads from others to increase their trust (therefore, the rank).

Defining the number of established links as the node *effort*, we want to study the relation between rank and effort, in order to know how much effort is required for a given rank. The proposed heuristic and the rank-effort relationship are examined in networks with a high number of nodes and different topologies - random and scale-free - also in presence of communities in order to establish to what extent the topology affects the attachment process. Our study shows that the behavior is very similar for all topologies although the number of steps (effort) needed to get the best rank changes according to the topology considered. Moreover, we noted that achieving a good rank is often much less expensive than getting the best, thus the heuristic is useful to choose the best tradeoff between a target rank and the related effort, for instance reaching the 20th position with 10 links is much better than being the first but after having connected to 1000 links.

This work continues the ongoing research on the best attachment strategy and trust dynamics we are working on [6, 7]. In section 2 we outline the trust network model and the definitions, whereas in section 3 we show the results of simulations, providing concluding remarks and future works in section 4.

2 The Trust Network: Model and Definitions

In the following, we briefly provide some terms and definitions to describe the context where our investigation is carried out; more details can be found in [6, 7].

The trust network model comes from the one widely accepted in literature [8, 9], i.e. it is modeled as a directed graph where the nodes (N) are persons and labeled arcs (E) represent trust relationships with a measure ($\mathcal{L}$) of the trust value according to a given metric; here we used EigenTrust [10] simple and efficient as a global trust metric. Trust values are used to rank nodes in a descending order, i.e. the more trusted the node, the better its rank. We also suppose that network nodes are all honest (no one tries to subvert others' trust values).

Finally, we define the *effort* as the cost the node X bears to persuade another node Y to trust it; once Y trust X, a link between them is established, thus the effort here is simply the number of links X establishes within the trust network. For the sake of simplicity before the new node X joins the network, all existing arcs are labeled with 1.0 as *local* trust value (the global value comes from EigenTrust and falls in

the range [0,1]). We choose this setting so the distribution of trust values does not affect our simulation results.

Considering the joining process, we adopt a pure random node selection attachment strategy since we want to focus only on the effects of the connectivity pattern of a network. Therefore, at every step the *target* node (i.e. the new node X joining the network) is connected by an in-link with a node chosen at random among those of the network. The process is stopped when the target node became the highest-rank node in the sense of the EigenTrust metric.

3 Simulation Results

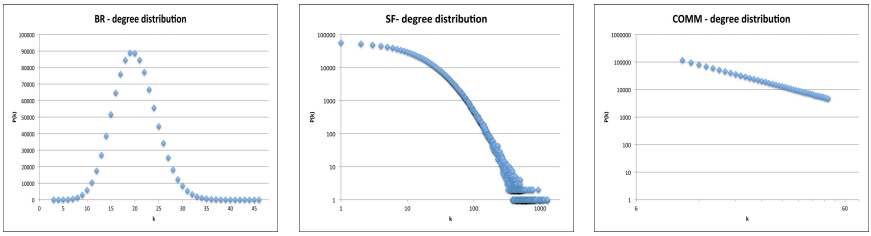
To study the correlation between network attachment and topologies, we conduct a set of experiments. Using the attachment strategy described before, we explore the trend of the target node rank on different network topologies. In particular, we consider Bernoulli random networks (BR), scale–free networks and random networks with a well defined community structure. A random BR network is generated by connecting nodes with a given probability p. The obtained network exhibit a binomial (Bernoulli) degree distribution [11]. Scale–free network (SF) [12] is a network whose degree distribution follows a power law, i.e. the fraction $P(k)$ of nodes having degree k goes as $P(k) \sim k^{-\gamma}$, where γ is typically in the range $2 < \gamma < 3$. A scale–free network is characterized by the presence of *hub* nodes, i.e. with a degree that is much higher than the average. Hubs often play an important role in a network and can be used as a key element to explain many behavioural patterns a network can exhibit. Finally, we consider a class of random networks with a given community structure (COMM in the following), generated via the benchmark proposed in [13].

Simulations have been performed by using a relatively small BR, SF and COMM networks of 2000 nodes each. Results are then confirmed by repeating the same simulations on networks with 1 million (1M) of nodes. Fig. 1 reports the main topological info of such networks, whereas in fig. 2 the degree distributions of 1M networks are shown.

In fig. 3 the results of our first experiment are reported. The rank of the target node is shown as a function of the number of the in-links in a 2K nodes networks.

name	#nodes	#links	average degree
BR	2000	39715	39.715
SF	2000	39195	39.195
COMM	2000	38838	38.84
BR_1M	1000000	19999121	39.99
SF_1M	1000000	19990188	39.98
COMM_1M	1000000	19568045	39.14

Fig. 1 Networks topological parameters



(a) (b) (c)

Fig. 2 Degree distribution for BR, SF and COMM networks

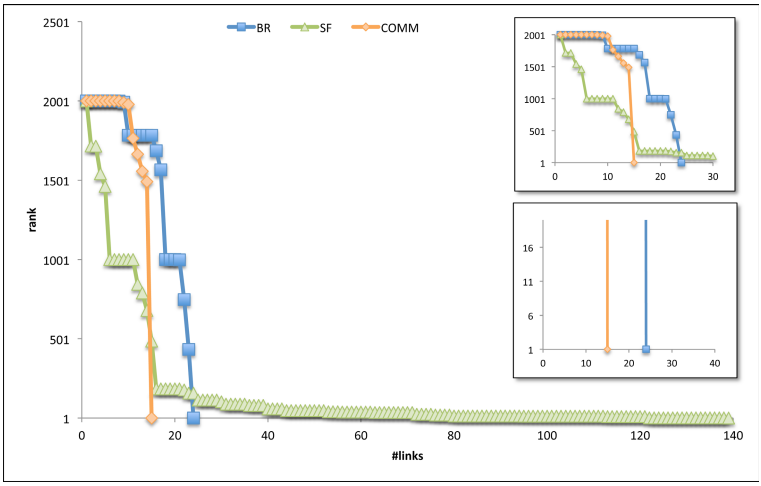


Fig. 3 Rank for BR, SF anc COMM networks - 2000 nodes

Several considerations arise; first of all, the number of in-links (or steps) to reach the top rank position heavily depends on the kind of network. In fact, while COMM and BR networks reach the first position in about 20 steps (figure lower inset), the SF network needs more than 130 steps! Conversely, the SF network exhibits a more rapid dynamics in the first part of the graph. With a few in-links, the target node gains a better position in a SF network than in BR or COMM networks. For instance, with only 10 links (figure upper inset) the target node reaches the position 1001 in the SF network, while it is placed in position 1784 and 1979 in BR and COMM networks respectively. This behaviour can be justified by considering the degree distribution of the networks used in our experiments. A SF network is characterised by the presence of hubs, but the majority of nodes have a very low degree. In this condition, the target node can reach rapidly a good positioning even with a few links. Though, due to the presence of hubs it is not so easy to reach the highest rank and a very high number of in-links are necessary. The BR network behaves in a different way, mainly because of the absence of hubs. In BR network the rank of the target node begins to improve

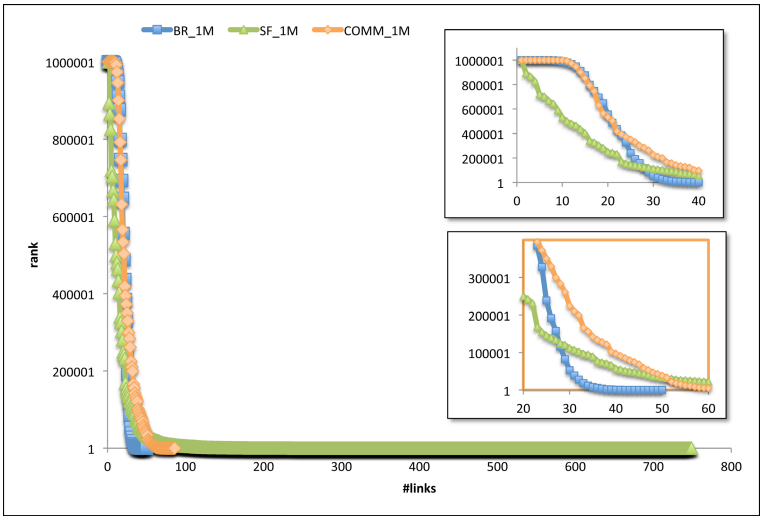


Fig. 4 Rank for BR, SF anc COMM networks - 1M nodes

when the number of in-links becomes more than 15, corresponding to the central part of the BR degree distribution curve. Again, the target node has a chance to improve its rank only when the number of in-links exceeds the average nodes degree of the network. In COMM network the behavioural pattern is similar to that of BR network, with a little improvement in the number of steps to reach the rank number 1. This is probably due to the presence of a community structure that impacts on the dynamic of the random walker underlying the trust metric we used. In fact, it is known that a random walker tends to remain trapped in a community, since nodes forming a community are connected one each other more than with the rest of the network. To confirm the considerations made so far, we performed the same experiments with 1 million nodes networks (see fig. 1). Apart from the number of steps need to reach the top rank, results shown in fig. 4 are quite similar to those in fig. 3 for the 2K nodes networks.

In conclusion, since many real world networks are SF, included most trust networks, we can try to define a set of principles an attachment strategy should follow. Focusing on SF networks, the first consideration is that it is possible to achieve a good rank in a few steps (i.e. little effort). This can be useful, for instance consider a company that wants to increase its visibility on the marketplace and plans both a target position and the corresponding capital investment it has to support over years. Referring to this example, our results imply that the company can reach an acceptable market position somehow limiting the effort (saving money). On the other hand, results also show that the effort for top rank positioning is very high, so if the company plans to become the first in the market this is possible but requires many steps, i.e. a lot of effort (money).

4 Conclusions

In this paper we studied how network topology is linked to the attachment problem in trust networks. In our study we adopt a simple random node selection attachment strategy. Our results are particular interesting in the case of scale–free networks, where we discovered that a very good rank is achievable in a very few steps, while a top rank positioning require a very high effort. Future works will concentrate on defining (1) a general recommendation to choose the best tradeoff between a target rank and the related effort and (2) a better attachment strategy to exploit more properties of scale–free networks.

Acknowledgements. This work was developed under the project "PON04a2_C (PON 2007-2013) - SMART HEALTH - CLUSTER OSDH - SMART FSE - STAYWELL" carried out by University of Catania.

References

1. Golbeck, J.: The dynamics of web-based social networks: Membership, relationships, and change. First Monday 12(11) (2007)
2. Artz, D., Gil, Y.: A survey of trust in computer science and the semantic web. Web Semantics: Science, Services and Agents on the World Wide Web 5(2), 58–71 (2007)
3. Briggs, P.: The evolution of trust. In: Wakeman, I., Gudes, E., Jensen, C.D., Crampton, J. (eds.) Trust Management V. IFIP AICT, vol. 358, pp. 13–16. Springer, Heidelberg (2011)
4. Nepal, S., Sherchan, W., Paris, C.: Building trust communities using social trust. In: Ardissono, L., Kuflik, T. (eds.) UMAP Workshops 2011. LNCS, vol. 7138, pp. 243–255. Springer, Heidelberg (2012)
5. Salam, A., Iyer, L., Palvia, P., Singh, R.: Trust in e-commerce. Commun. ACM 48(2), 72–77 (2005)
6. Carchiolo, V., Longheu, A., Malgeri, M., Mangioni, G.: Users attachment in trust networks: Reputation vs. effort. Int. J. Bio-Inspired Comput. 5(4), 199–209 (2013)
7. Carchiolo, V., Longheu, A., Malgeri, M., Mangioni, G.: A heuristic to explore trust networks dynamics. In: Zavoral, F., Jung, J.J., Badica, C. (eds.) IDC 2013. SCI, vol. 511, pp. 67–76. Springer, Heidelberg (2013)
8. Golbeck, J.A.: Computing and applying trust in web-based social networks. PhD thesis, College Park, MD, USA (2005); Chair-Hendler, J.
9. Walter, F.E., Battiston, S., Schweitzer, F.: A model of a trust-based recommendation system on a social network. JAAMAS 16, 57 (2008)
10. Sepandar, D., Kamvar, M.T., Schlosser, H.G.M.: The eigentrust algorithm for reputation management in P2P networks. In: proceedings of the Twelfth International World Wide Web Conference 2003 (2003)
11. Albert, R., Barabasi, A.L.: Statistical mechanics of complex networks. Reviews of Modern Physics 74, 47 (2002)
12. Barabasi, A.L., Albert, R.: Emergence of scaling in random networks. Science 286, 509 (1999)
13. Lancichinetti, A., Fortunato, S.: Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. Phys. Rev. E 80, 016118 (2009)

Part XII
MASTS'2014 Papers

Data Fusion in a Multi Agent System for Person Detection and Tracking in an Intelligent Room

Matei Chiperi, Mihai Trascau, Irina Mocanu, and Adina Magda Florea

Abstract. The main components of a supervising system is detecting and tracking of the supervised person in an intelligent room. This paper presents architecture for a non-intrusive multi-agent system for person detection and tracking. The main objective of this system is to offer continuity over the user's movement, as it can be controlled in such a way so as to keep the user inside the frame for most of the time. The proposed architecture will integrate different types of sensors: multiple Kinect sensors and a PTZ camera, in order to minimize the drawbacks of using only one type of sensor. For example field of view provided by Kinect sensor is not wide enough to cover the entire room. Also the PTZ camera is not able to detect and track a person in case of different special situations, such as the person is sitting or it is under the camera. Furthermore Kinect sensors will help the PTZ camera to control the camera's orientation. Person detection and tracking is performed using computer vision techniques applied to RGB images. The system is designed over the existing platform AmI-Platform and is partially evaluated in the AmI-Lab laboratory from the University Politehnica of Bucharest (UPB).

1 Introduction

Ambient Intelligence (AmI) is an actively studied topic in the world of Artificial Intelligence. Its vision is to create a smart environment, sensitive and responsive to the presence of people and their activities. While there are various descriptions for such an environment, there are some key points common to all the opinions expressed: the user is the central point of such an environment; all the gadgets, technologies

Matei Chiperi · Mihai Trascau · Irina Mocanu · Adina Magda Florea
University "Politehnica" of Bucharest, Computer Science Department,
Splaiul Independentei 313, 060042 Bucharest, Romania
e-mail: {matei.chiperi,mihai.trascau}@gmail.com,
{irina.mocanu,adina.florea}@cs.pub.ro

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_40

and automation features must be of aid, with minimum configuration and operation effort.

Starting from this perspective, detecting and tracking a person in an intelligent room are the main objectives of this paper. Having a pan-tilt-zoom camera (PTZ) and a number of Kinect sensors in an intelligent room, we propose a non-intrusive multi-agent system that is able to detect and track a person in real-time. The particularities of the problem involve the variability of a person's pose inside a room (standing, sitting, lying), the cluttered background, dead angles (there might be portions of the room which are not covered by the tracking or image systems) or uncomfortable angles in which the detection has to be performed (a person might be sitting right under the camera or might be keeping his hands crossed in front of the Kinect sensor). Agents collect data from multiple and different types of sensors: multiple Kinect sensors and a PTZ camera. We combine these two types of sensors in order to obtain continuity over the user's movement, as it can be controlled in such a way so as to keep the user inside the frame for most of the time. Also Kinect sensors will help the PTZ camera to control the camera's orientation. The agents are implemented on top of AmI-Platform, a framework for integrating sensors and actuator in an ambient intelligent environment. Performance and validation tests are performed in AmI-Lab at UPB. The proposed system considers only one person in the supervised room. Also the system doesn't consider the privacy implications of having supervising cameras in the person's home.

The rest of the paper is organized as follows. Section 2 describes related work about existing ambient intelligent systems and different methods for person detection, and tracking. Section 3 describes the proposed multi-agent system used for person detection and tracking. Section 4 presents the current evaluation of the proposed system. Conclusions and future works are listed in Section 5.

2 Related Work – Computer Vision Based Solutions for Person Detection

Two major approaches dominate the field of person detection. One uses a parts-based approach, in which the aim is to detect separate body parts and reconstruct an articulated body, in a plausible configuration. The paper [3] uses a pictorial structure approach: an object is represented by parts connected with springs. The other main approach uses a sliding detection window. Several of the most popular person detection algorithms use this technique. A sliding detection window approach to detect pedestrians is used in [4] - the edge information from the window is compared to an exemplar dataset using chamfer distance. A new object detection approach using a sliding window, based on evaluating well-normalized local histograms of image gradient orientations in a dense grid is proposed in [1]. The paper introduced the concept of Histogram of Oriented Gradients (HOG), which represents a set of feature descriptors for an image.

The method is based on the idea that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. The descriptors are obtained by dividing the image into small connected regions (cells) and for each cell compiling a histogram of gradient directions. The approach described in [1] uses detection windows of 64x128 pixels. Each window is divided into 8x8 pixel cells, and groups of 2x2 cells are grouped into blocks. The blocks are considered in a sliding fashion, allowing overlapping. Each cell provides a 9-bin histogram of oriented gradients, and each block contains a concatenated vector of all its cells' bins. A detection window is represented by 7x15 blocks, resulting a feature vector of 3780 elements. The presented approach uses a SVM classifier. In [9] the HOG approach is modified by using variable-sized and variable-ratio blocks and an AdaBoost classification process, allowing a significant improvement in processing time. [8] added Local Binary Patterns (LPB) as descriptor features, besides HOG. Also, they provide a solution for detecting partially occluded persons, based on the SVM's answer to negative portions of the image. In [6] it is employed an integral HOG and oriented LBP in order to obtain a speedup, and also added a Kalman filtering step for tracking persons in successive video frames. In [2] it is extended the concept of HOG person detector to video streams: a HOG appearance descriptors is computed from the current image frame, and another HOG descriptor is computed from the optical flow differences between successive video frames. This approach obtained better results compared to the ones from [1].

3 System Description

We develop a system capable of detecting and tracking one person. From a theoretical approach, a single camera might have limited performance in the process of tracking a person and detecting its position. This leads to the inclusion of a pan-tilt-zoom camera combined with several Kinect sensors. The limitations imposed by the Kinect's technical specifications are a major limiter for the performance of a tracking system deployed inside a large room. In order to overcome these limitations, we have coupled more Kinect sensors spread around the room and aggregate the data in order to obtain a global picture of a person's position and movements. One obstacle at this step will be the interference between the Kinect sensors, as they use infrared sensors operating at the same frequency. The infrared waves transmitted by two different Kinect sensors can interfere, leading them to derive erroneous data. The system is designed as a multi-agent system. It is developed based on the infrastructure of AmI-Lab from the UPB using the existing AmI-Platform [5].

3.1 AmI-Lab

The AmI-Lab at the UPB is a room of 8.5 m x 4.5 m, equipped with various tracking sensors. The available tracking system is composed of accessible, off-the-shelf sensing components: (a) 9 Microsoft Kinect sensors (K1 âĂŞ K9); these devices contain an RGB camera and an infrared depth sensing device - they are capable to deliver RGB images and the associated depth information at 30 frames per second, the range field being 0.7 to 6 meters, according to the official specifications; (b) one pan-tilt-zoom Samsung H.264 Network PTZ Dome Camera. This device allows focusing a person and track his movements in the environment, at a 360° angle; it is capable of optical zooming (up to 12X), and it can deliver clear RGB streaming of up to 30 frames per second; its high levels of freedom allow tracking an individual in every spot of the room, making it a good complement for the Kinect sensors. The Kinect sensors are placed as evenly spaced as possible on the room's edges, while the PTZ camera is placed on the ceiling, in the center of the room.

3.2 AmI-Platform

The sensing infrastructure of the AmI-Platform is built on top of a number of Microsoft Kinect sensors and Arduino boards, from which data is fetched using either a wired USB connection (for high-bandwidth measurements such as frames), or a wireless 802.11/b connection (for discrete measurements such as the output of an infrared proximity sensor). The software that processes this data is organized as a pipeline made up of individual PDUs (processing data units) communicating through a system of distributed queues, as described in [5].

3.3 The Mult-agent System for AmI-Platform

There are several problems with this approach used in the AmI-Platform which we will try to solve by integrating individual agents of a multi-agent system which will coordinate analysis and predictions from the Kinects and the PTZ camera. First, the FOV (Field of View) that the Kinects provide is not wide enough to cover the entire room. Areas that are especially prone to this problem are the corners of the rooms, the immediate vicinities of the Kinects (somewhere under 1.5 meters), and at relatively large distances (over 4 meters). These "blind spots" disrupt the continuity of the user's trajectory. Moreover, there is a latency in the person detection and skeleton extraction by the Kinect, which further hinders continuity. The immediate solution for this problem was to increase the number of Kinect sensors in the room. However, this solution both increases the cost of the system, making it financially unfeasible for deployment in a real house of office. Also, by increasing the number of Kinects the problem of interference between their lasers arises which is due to the Kinects

operating at the same frequency. This problem is recurrent for projects that employ Kinect sensors, and is admittedly a hard one to solve. Other issues that proves problematic when using the Kinect is the fact that having both depth data and RGB data recorded would result in large amounts of information being pushed to the system. This results in tighter constraints on bandwidth, storage space and computational power, which affects the scalability of the system and imposes very large requirements for computational resources.

Our alternative was to use a PTZ camera that will be used in conjunction with the Kinect sensors. This camera would help in offering frame continuity over the user's movement, as it can be controlled in such a way so as to keep the user inside the frame for most of the time. The multi-agent system we envision would contain different types of agents that need to cooperate in order to satisfy the goal of detecting and tracking the user.

The present solution involves the use of a multi-agent system (MAS). The agents are organized based on their role, and we have defined two major layers: (a) Data Acquisition & Processing; and (b) Aggregation & Decision Making. The agents responsible for data acquisition and processing will be the ones directly connected to the sensors, and they are known as sensing agents. For our purposes, we will have 3 agents each in charge with the data from a Kinect sensor and one agent responsible for the PTZ camera. Another agent, known as manager agent, has the responsibility to aggregate data from the sensing agents and act upon their estimations of the user's position and their prediction for the future position. Based on its decisions, the user's trajectory can be inferred with higher precision. Considering that the PTZ agent needs to control the camera's orientation in order to maintain the user inside the frame, the manager agent can help the PTZ agent in his decision. Also, the computer vision algorithms used for person detection by the PTZ agent have their performance reduced in complicated situations (the user is sitting, the user is right under the camera, the user is obscured by some larger objects, or the user appears incomplete in the frame). Therefore, the manager agent can indicate when requested by the PTZ agent, a globally estimated position of the user by employing the estimations/predictions made by the Kinect agents.

3.4 PTZ Person Detection and Tracking Solution

The most feasible approach for detecting a person seems to be the sliding window technique, in which features are extracted from a portion of the image and confronted with a classifier. We have opted for this technique due to the given circumstances of the environment: a camera capable of capturing a subject in all the corners of a room, but from different angles. Due to this constraint, the parts-based approach seems unfeasible, as not all the body parts will be visible under certain viewing angles. This constraint represents a challenge and a novel problem.

The used features include appearance descriptors and also motion descriptors, similarly to [2]. We consider the motion descriptors helpful, as the static appearance

descriptors might prove to be insufficient: different person poses taken from different angles will vary the resulting descriptors significantly. An ideal situation would be a set of training images for most of the viewing angles, both for people standing upright, bending, sitting etc., but this is physically impossible in terms of necessary possible situations and computational resources.

Considering the problems described in the previous paragraph, the classification component needs to be a very general and flexible one. It must correctly classify both people standing and sitting, people seen from full profile or from above etc. These are very hard requirements to impose for a single classifier.

A final component necessary for the tracking part consists in moving the camera in order to follow the subject. We conceive to use motion information extracted from 2 or 3 successive images in order to predict the movement direction of the person. A coarse prediction of the subject's next step allows the movement of the PTZ camera in order to maintain the person in focus. Fig. 1 gives an overview of the general architecture considered for the system.

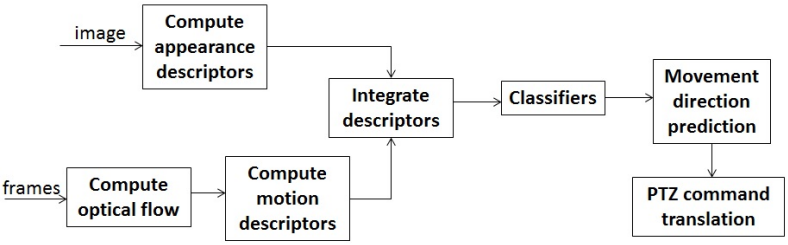


Fig. 1 PTZ person detection and tracking system architecture

A fundamental decision lies in the choice of the image descriptors used. Combining both static appearance descriptors (operating on a single picture) and motion descriptors (operating on consecutive pictures) is a beneficial approach, as proved by various experiments [1], [7]. As a first approach, the decision is to use Histogram of Oriented Gradients grids as a part of the static-image part of the descriptor [1]. In terms of motion descriptors, we project the usage of a similar approach to the one described by [2]. These motion features are expected to improve the results, but also to increase the final feature dimension.

4 Evaluation of the PTZ Person Detection Performance

4.1 HOG Static Descriptors

As an initial implementation step, we tested a readily available solution for person detection and compare the results with the future solutions. In the light of the outlined

approach, a test was performed using the HOG person detector described by [1] and implemented in the OpenCV library. The HOG person detector implemented by the OpenCV framework uses a detection window of 128x64 pixels and is trained on the INRIA dataset. The detection process was applied using multiple scaled versions of the image, on a scale step of 1.05. The HOG parameters used are the following: cell size of 8x8 pixels, block stride of 8x8 pixels, block size of 16x16 pixels (2x2 cells). The image set on which the test was performed is a set of frames acquired by the PTZ camera from a fixed position, with the optical zoom set to 1x. The images have a resolution of 640x480 pixels. The preliminary experiments showed the necessity of tuning the detection process, hence the following heuristics were added to the person detection process: a detected window should not be smaller than a specified threshold (experimentally found to be 256x512), in order to filter out false positives and keep the windows which most probably capture a person. Another decision taken was to discard the resulting positive windows which present a confidence lower than 1. Due to the experiment conditions, in which the camera is fixed and people pass in and out of the captured space (the camera does not follow a subject), it was expected that the system would provide average results for images of persons captured in difficult conditions. For instance, people partially occluded (entering or exiting through a door or exiting the captured space - Fig. 2(d), persons captured in various positions (bending, sitting on a chair - Fig. 2(a), moving their arms) or persons being captured under poor light conditions were not extremely accurately detected by the person detection system.

(a) Sitting person (b) Standing person detected (c) Close angle - person not detected (d) Partially occluded person

Fig. 2 Results obtained by the static HOG person detector in various situations

A problem identified during the experiments was the deformation of the image caused by the camera itself. Since the experiment is using a PTZ camera, fixed on the ceiling of the room, the central area of the captured image is in the desired shape. The areas placed towards to edges of the image present a slight deformation, which makes the usage of a detection window more difficult.

A second difficulty identified experimentally revealed that towards the center of the room (as the subject gets closer under the camera), the detection system will perform poorly, due to the angle under which the image was captured (Fig. 2(c)). The persons will look less like normal stances of standing people, a new set of training data being required for these cases. A countermeasure for this scenario is provided by the Kinect sensors, which cover the central parts of the room. We conducted two sets of experiments: one for a more relaxed case, in which we expected the system to

detect fully visible persons, captured in the center of the image, under wider angles, and a second one for a more general approach, in which we penalized the system for not detecting people placed towards the edges of the image, partially occluded (but visible enough to be easily recognized by a human). In the first case, we analyzed 561 random images, obtaining an accuracy of 83.7%, a precision of 96.6% and a recall of 70.4%. In the second case, 358 random images were tested, for an accuracy of 72.9%, a precision of 96.1% and a recall of 69.2%.

4.2 Adding Motion Flow Descriptors

As a result of the limited expected performance of the person detector implemented using static HOG features, we decided to add motion descriptors features to the classification process. Influenced by [2], we used the static HOG descriptors in conjunction with the HOG descriptors of the motion flow image resulted from two successive frames. The Farneback optical flow method was chosen for computing the optical flow image between two consecutive static frames. For each detection window, there are two HOG descriptors: one for the image of the current frame (the static features) and one for the optical flow image corresponding to the current frame and the previous one (the motion features). The concatenation of the two HOG descriptors represents the final feature set for the window.

Because a custom feature set was computed, a new classification process was needed. For this, we manually selected 762 image patches of people from recorded video streams for the positive set. Each patch was saved in association with its motion flow image (Fig. 3(a), Fig. 3(b)). In order to increase the number of positive images, we vertically flipped the images, doubling the total number of positive images. For the negative samples, we randomly selected a number of image patches not containing people. In order to increase the number of negative samples, we also took frames not containing people and added all the possible windows to the negative set (a variation of the hard negatives described by [2]). Thus, we obtained a total of 4400 negative samples (Fig. 3(c), Fig. 3(d)). For the computation of the optical flow, we used the Farneback optical flow implementation available in OpenCV. For classification, we used a Python port of the LibSVM available in the Scikit-Learn library, training a linear SVM; the window size used was of 128x64 pixels, with a total number of 7560 features per window. For testing, a number of captured video streams was used - each frame was scaled down such that a 128x64 pixels window would enclose a man silhouette. Each frame was associated with its optical flow image, computed using the preceding frame (the first frame is not taken into consideration, as it doesn't have a predecessor). A sliding window was run along the frame at 16 pixels steps. For each window, the set of static and motion features was computed and it was served to the classifier. The probability of the window belonging to the positive class was the discriminant in choosing the best window for the frame.

The results obtained were significantly better than the ones obtained by using a static features person detector. We ran the detection process on two test sets, each

(a) Positive image (b) Positive motion flow (c) Negative image (d) Negative motion flow

Fig. 3 Samples of positive and negative windows and their associated motion flows

containing over 400 frames. Thare are used two sets. The two sets are different in conditions - the first set contains a static scene, in which the camera does not move and the persons to detect are in relatively clear standing positions. The second set contains passages of camera movement, in which the scene changes, and the people presented are also placed in less obvious positions - sitting, partially occluded by furniture pieces. The camera movement and the changes of the background are among the causes for the higher number of false positives. The training of a custom classifier proved beneficial for the non-standard positions of the persons: the detector managed to detect instances of sitting persons, not fully visible (partially hidden by a door/exiting the frame), or being close under the camera. In the first case, we analyzed 420 images, obtaining an accuracy of 88.5%, 33 false positives and 15 false negatives. In the second case, 512 images were tested, for an accuracy of 80%, 90 false positives and 10 false negatives.

Thus, a large number of negative examples can significantly improve the performance of the classifier. A second conclusion highlights the beneficial effect of the addition of the optical flow features, at the cost of a higher processing time for each frame.

As a downside of this approach, the processing time for each frame is very high, a few orders of magnitude higher than the first approach of static descriptors. It must be mentioned that we haven't approached the problem of optimizing the execution time. Also, for this approach, we used a simple implementation of the sliding window mechanism, contrasting with the optimized solution offered by the OpenCV library, employed in the first experiment. Another solution would be the usage of a faster, less accurate implementation of the HOG descriptors.

5 Conclusions and Future Work

The paper presents a prototype system for person detection and tracking in an intelligent environment. The system is based on the multi agent architecture. The multi agent system collects all the necessary information from different data sources: Kinect sensors and PTZ camera in order to improve person detection and tracking. These sensors will help in offering frame continuity over the user's movement, as it can be controlled in such a way so as to keep the user inside the frame for most of the time. Also Kinect sensors contribute in controlling the PTZ camera's orientation. The person detection is made by using static and dynamic HOG descriptors applied to RGB images. For the moment tracking a person is obtained by person detection. As future work the person detection part will be improved by using a cascade of classifiers, each set dealing with certain poses or angles. Also the tracking part will be made by using the Kalman-ïňĄltering algorithm. Regarding the implemented algorithm, we intend to optimize of the computing time needed for the calculation of the static and dynamic HOG descriptors. One solution would be the usage of a faster, less accurate HOG descriptor implementation, combined with an optimized sliding window mechanism implementation.

Acknowledgements. This work is supported by the Sectoral Operational Programme Human Resources Development (SOP HRD), financed from the European Social Fund and the Romanian Government under the contract number POSDRU/159/1.5/S/137390/.

References

1. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1. IEEE (2005)
2. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 428–441. Springer, Heidelberg (2006)
3. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. International Journal of Computer Vision 61(1), 55–79 (2005)
4. Gavrila, D.M., Philomin, V.: Real-time object detection for smart vehicles. In: The Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 1. IEEE (1999)
5. Ismail, A., Florea, M.: Multimodal indoor tracking of a single elder in an aal environment. Ambient Intelligence – Software and Applications Advances in Intelligent Systems 219, 137–145 (2013)
6. Ma, Y., Chen, X., Chen, G.: Pedestrian Detection and Tracking Using HOG and Oriented-LBP Features. In: Altman, E., Shi, W. (eds.) NPC 2011. LNCS, vol. 6985, pp. 176–184. Springer, Heidelberg (2011)
7. Viola, P., Jones, M.D., Snow, D.: Detecting pedestrians using patterns of motion and appearance. International Journal of Computer Vision 63(2), 153–161 (2005)
8. Wang, X., Han, T.X., Yan, S.: An hog-lbp human detector with partial occlusion handling. In: IEEE 12th International Conference on Computer Vision. IEEE (2009)
9. Zhu, Q., Avidan, S., Yeh, M.-C., Cheng, K.-T.: Fast human detection using a cascade of histograms of oriented gradients. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2. IEEE (2006)

Emergence of Norms in Multi-agent Societies: Influence of Population Size and Topology

Marius-Tudor Benea and Mihai Trăscău

Abstract. This work studies the problem of the emergence of norms in multi-agent systems. Based on a socially interactive game environment for the "rules of the road" problem, of choosing a side of the road to drive on, a series of simulations are run in order to highlight the importance of the topology of the network underlying the multi-agent system and the influence of the population size. The simulation environment is optimized to run simulations of large multi-agent systems and the network topologies studied are: strongly regular, complete, random, random regular, small-world and scale free.

1 Introduction

The role of norms in a society is important, as they are the result of a complex process of emergence, based on the social interactions between the members of that society. For this reason they tend to be generally accepted by all the members of the society, because this property of norms makes them better suit the needs of the population, as opposed to the top-down mechanism of enforcement of the laws. Moreover, fewer resources are required from a central authority for their creation and assimilation by the society. However, there is still a cost, the lack of precision. For this reason, it becomes vital to understand the mechanisms behind the emergence of norms.

Marius-Tudor Benea
LIP6, University Pierre and Marie Curie, France
e-mail: marius-tudor.benea@lip6.fr

Mihai Trăscău · Marius-Tudor Benea
Computer Science Department, University Politehnica of Bucharest, Romania
e-mail: mihai.trascau@cs.pub.ro

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_41

Many aspects influence the emergence of norms in multi-agent systems, like the underlying network structure, the representation of norms and the propagation mechanisms, the types of social interactions and their frequency, or the normative strategies of the agents. In this article we intend to study the way in which the topology of the network of interactions behind a multi-agent system and the size of the population influence the norm emergence speed. For this purpose we use the scenario of choosing the side of the road to drive on.

This work is part of a series of studies on the emergence of norms in multi-agent societies, and it extends and improves the works done in [1,9]. The simulation environment is optimized for running simulations of large multi-agent systems and the network topologies studied are: strongly regular, complete, random, random regular, small-world and the Barábasi extended model of scale free graphs.

2 Related Work

With its decentralized behavior-shaping effect, the emergence of norms is a potentially scalable, adaptive control mechanism. While works like [2,3,6–8] study it too, they do it from different perspectives. A comprehensive study of norms is offered in [5], based on a proposed life-cycle model of norms.

None of the works mentioned above study in depth the influence of the topologies on emergence of norms, nor the influence of the population size. Mukherjee et al., in [2, 3] treat the problem of the structure of the society just a little, through what they call "spatially constrained interactions", taking the form of agent neighborhoods. Josep Pujol's PhD thesis, [4], is another example. We extend his study with some other topologies, *complete graphs* and *random regular graphs*, both having properties worth considering, like the high connectivity between the nodes or a constant degree of nodes and a random component.

A framework for the study of the mechanisms of norm emergence, including the study of the influence of network topologies, was created as part of the works described in [1,9]. The current work extends [1] by averaging the results over a larger number of simulations, using an improved simulation environment, and for a different scenario. Also, a strange behavior of the strongly regular graphs observed in [1] was further studied. In [9] the study of the influence of topologies was postponed, due to the lack of efficiency of the simulation module of the framework. We improved the framework and reconsidered the study, for a different scenario, and we also made use of the advantages offered by the new framework to study the influence of the population size.

3 Simulation Model and Framework

The scenario used is the one of the side of the road to choose to drive on (left or right), described in [2]. If the agents choose the same action, they are rewarded with 1, otherwise they will get -1. During each epoch, all the agents play at least once.

The simulation platform used in [1, 9] was modified. We decided to quit using JADE for the simulations. This way we avoided some unnecessary computations and waiting times, not needed for simulations of large multi-agent systems, for which a much simpler approach is better. Thus, the performance of the environment was significantly increased, rewriting it from scratch, using only basic Java. Except for this difference, the behavior of the platform is mainly the same. We use the same simulation architecture, the existing QAgent agent model (the same for all the agents) and the same network generation module as before. The topologies used are also the same, namely: complete, strongly regular, random - the Erdos-Rényi model, random regular, small-world - the Watts-Strogatz model, scale free - the extended Barábasi model.

4 The Influence of the Population Size

The first results for this particular problem are shown in Table 1, column 3. For the population size of 2000 we generated 5 graphs for each topology, except for the "complete" case for which one graph is enough, and ran the simulation 10 times for each generated graph. For the other sizes these values were 15 and 100, respectively, with the same exception. The mean degree of the nodes of the last two topologies is around 14.

We can see that the influence of the population size on the emergence of the norms is not very significant, so the norm emergence process scales very well. For example, for the most representative example of the complete graph, the number of epochs, for a population 10 times greater, increases by 24% of the first emergence time. Even though this property is not obvious for the other two cases, we observed that the cause was a small number of agents who remained neutral (agents that can choose different policies, depending on the role, or that have equal utilities for one role) for a considerable number of epochs. To deal with this problem, we repeated the simulation for the emergence thresholds of 99% and 95% (Table 1, columns 4-5). We can see significant improvements in this case, especially for the Barábasi extended model graphs.

Table 1 Number of epochs to emergence for the thresholds: 100%, 99% and 95%

Network type	Parameters	No. of epochs / threshold		
		100 %	99 %	95 %
Complete	N = 200	50	43	40
Complete	N = 500	54	48	42
Complete	N = 1000	59	54	45
Complete	N = 2000	62	55	59
Random	N = 200, m = 1400	75	79	72
Random	N = 500, m = 3500	94	87	77
Random	N = 1000, m = 7000	125	90	91
Random	N = 2000, m = 14000	122	98	87
Barabási www	N = 200, m_0 = 7, m = 7, p = q = 0.4125	92	70	55
Barabási www	N = 500, m_0 = 7, m = 7, p = q = 0.4125	141	90	66
Barabási www	N = 1000, m_0 = 7, m = 7, p = q = 0.4125	219	104	79
Barabási www	N = 2000, m_0 = 7, m = 7, p = q = 0.4125	269	122	87

5 The Influence of the Topologies

To find out the role of network topologies in the emergence process, we made an average of a large number of runs, for the results obtained to be as accurate as possible. Thus, we ran up to ten thousand simulations for one network type: we generated 100 graphs for each topology that includes a random component (all except for complete and strongly regular, for which one graph was enough) and we ran 100 simulations for each of them. The average degree of a node is close to 14, except the complete graph. The emergence threshold was set to 95%. The results, together with the parameters used to generate the networks, can be seen in Table 2.

We see that the fastest topology is the complete graph. This is caused by the high connectivity of the network which causes the changes to propagate very fast. However, this topology is an unrealistic one. From the next two results we conclude that the distribution of the degrees of the nodes contributes significantly to the emergence speed. Random regular networks, in which the degree of each node is constant, are better than the random ones. This is due to the fact that, in the random case, the agents placed in the nodes with a smaller degree convert slower to the norm of the society, because of their limited interactions. Thus, even if the random graphs have some nodes with a high degree, which proved to be an important property, as we could see for complete graphs, they do not substitute the disadvantage of the nodes with fewer neighbors.

The strange behavior of the strongly regular graphs, seen in [1], was observed again. Thus, the system hasn't converged to a single norm. The phenomenon is exemplified and can be watched a little bit closer in Fig. 1. Neighborhoods of agents with specific policies are determined to be created by the structure of the network.

Table 2 Number of epochs to emergence for different topologies; Threshold = 95 %

Network type	Parameters	Epochs
Complete	N = 200	39
Random	N = 200, m = 1400	62
Random regular	N = 200, k = 7	58
Strongly regular	N = 200, k = 7	∞
Small-world	N = 200, k = 7, p = 0.5	68
Small-world	N = 200, k = 7, p = 0.75	59
Barabási citation, $\gamma = 3$	N = 200, m_0 = 7, m = 7, p = q = 0	82
Barabási www, $\gamma = 2.1$	N = 200, m_0 = 7, m = 7, p = q = 0.4125	68
Barabási math, $\gamma = 2.5$	N = 200, m_0 = 7, m = 7, p = q = 0.2812	54

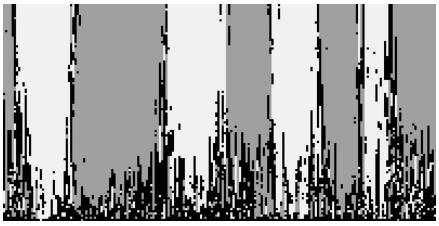


Fig. 1 Overview of the local norms creation process for strongly regular graphs; x axis = agents (1-200), y axis = epochs (1-100); Policies: black = neutral, grey = left, white = right

For the small-world graphs, we see that the one that is closer to a random graph (p=0.75) makes the system run faster. This is due to the fact that the graph generated with p=0.5 is closer to the structure of a strongly regular graphs, thus it has a higher potential to conduct to the emergence of local norms, as seen above. We also ran simulations for p=0.25, but in this case, for some simulations, our system ran for a large number of epochs until it reached a configuration similar to the one in Fig. 1. For the best case, small-world graphs were faster than the completely random ones, which shows the advantages of the small characteristic path lengths describing them. The complete and random regular topologies were faster, but they are also unrealistic. Moreover, the results of the random regular case are comparable.

The last discussion is about the scale free graphs. The slowest case is the citation one. This behavior is caused by the fact that there are some nodes, the ones created at the end, with a lower degree and here comes the problem discussed earlier, caused by the limited interactions. An improvement is observed in the last two cases. The case of Barábasi math is the fastest. So, the last added nodes communicate better now, when some links were rewired and some others were added. The results obtained in the Barabási www case highlight the importance of the preferential probabilities. In this case the structure created with these probabilities is destroyed while adding and rewiring links to the existing nodes in about 82% of the cases. Thus the graphs tend to

be closer to a random one, and even though they have more links they are slower than the math case. We conclude thus that the network hubs, created using preferential probabilities, have a very positive influence on the results, but taken alone they tend to negatively influence them.

6 Conclusions and Future Work

In this paper we studied the influence of the population size and of the topology on the emergence of norms in multi-agent systems. We studied this based on a “rules of the road" scenario of choosing the side of the road to drive in.

The distributed mechanism of norms creation has proved its importance and performance. Thus, an interesting fact that we could observe was that norms have the potential to emerge in comparable time intervals for a small network as well as for a big one, the norm emergence process scaling, thus, very well.

Considering the influence of the topology, we were able to show that the hubs of the scale-free networks and the property of small characteristic path lengths of the small-world networks, both found in the real world societies, positively influence the emergence of norms. We also saw how in the case of strongly regular graphs our system didn’t converge to a single norm, small regions with local norms being created, and we studied this phenomenon. We also observed the importance of the high connectivity of the networks and the drawback of nodes with a low degree.

As future plans, we intend to use the platform, which proved its utility, to conduct more studies for different and more complex scenarios. We also intend to extend the studied set of topologies.

Acknowledgements. This work is supported by the Sectoral Operational Programme Human Resources Development (SOP HRD), financed from the European Social Fund and the Romanian Government under the contract numbers POSDRU/159/1.5/S/137390 and POSDRU/159/1.5/S/132395.

References

1. Benea, M.T., Tărtăreanu, T.A., Trăscău, M.: Norm emergence in multi-agent systems based on social interactions. Computer Science Master Research 1(1), 12–24 (2011), http://csmr.cs.pub.ro/index.php/csmr/article/view/33
2. Mukherjee, P., Sen, S., Airiau, S.: Norm emergence in spatially constrained interactions. In: Working Notes of the Adaptive and Learning Agents Workshop at AAMAS, vol. 7 (2007)
3. Mukherjee, P., Sen, S., Airiau, S.: Norm emergence under constrained interactions in diverse societies. In: Proceedings of the AAMAS 2008, vol. 2, pp. 779–786 (2008)
4. Pujol, J.M.: Structure in artificial societes. Ph.D. thesis, Universitat Politecnica de Catalunya, Departament de Llenguatges i Sistemes Informatics, PhD Program: Artificial Intelligence, Barcelona (2006)

5. Savarimuthu, B.T.R., Cranefield, S.: A categorization of simulation works on norms. In: 2009 (Dagstuhl Seminar Proceedings 09121: Normative Multi-Agent Systems, Leibniz, Germany, March 15-20, pp. 39–58 (2009)
6. Savarimuthu, B.T.R., Purvis, M., Cranefield, S., Purvis, M.: How do norms emerge in multi-agent societies? mechanisms design. In: The Information Science Discussion Paper Series, Number 2007/01, pp. 1177–1455 (February 2007) ISSN 1177-455X
7. Savarimuthu, B.T.R., Purvis, M., Purvis, M.K., Cranefield, S.: Social norm emergence in virtual agent societies. In: Baldoni, M., Son, T.C., van Riemsdijk, M.B., Winikoff, M. (eds.) DALT 2008. LNCS (LNAI), vol. 5397, pp. 18–28. Springer, Heidelberg (2009)
8. Sen, S., Airiau, S.: Emergence of norms through social learning. In: IJCAI 2007: Proceedings of the 20th international Joint Conference on Artifical Intelligence, pp. 1507–1512 (2007)
9. Trăscău, M., Benea, M.-T., Tărtăreanu, T.A., Radu, Ş.: Emergence of norms in multi-agent societies: An ultimatum game case study. In: Badica, A., Trawinski, B., Nguyen, N.T. (eds.) Recent Developments in Computational Collective Intelligence. SCI, vol. 513, pp. 37–46. Springer, Heidelberg (2014)

From Intentions to Plans: A Contextual Planning Guidance

Ahmed-Chawki Chaouche, Amal El Fallah Seghrouchni,
Jean-Michel Ilié, and Djamel Eddine Saïdouni

Abstract. The proposal AgLOTOS algebraic language is dedicated to the specification of agent plans in ambient systems (AmI). From its two level specification, plans can be built automatically as a system of concurrent processes. In this context, we show how to achieve a powerful mechanism for a contextual guidance based on a specific and formal construction called Contextual Planning System (CPS). The CPS structure is used to propose an optimal plan preserving the consistency of the intentions.

1 Introduction

Ambient Intelligence (AmI) is the vision of ubiquitous electronic environment that is non-intrusive and proactive, when assisting people during various activities [5, 7]. For the design of such complex systems, MAS approaches offer interesting frameworks, since their agents are considered as intelligent, proactive and autonomous [4].

The major problem for AmI agent consists in recognizing its environmental contexts, including its locality and the discovery of other agents. In [2], it is shown how autonomous BDI agents [8] can evolve and move within an ambient environment,

Ahmed-Chawki Chaouche
LIP6 Laboratory, University Pierre and Marie Curie - Paris 6, 4 Place Jussieu, 75005 Paris, France
(in cotutelle with MISC Laboratory, University Constantine 2)
e-mail: ahmed.chaouche@lip6.fr

Amal El Fallah Seghrouchni · Jean-Michel Ilié
LIP6 Laboratory, University Pierre and Marie Curie - Paris 6, 4 Place Jussieu, 75005 Paris, France
e-mail: amal.elfallah@lip6.fr, jean-michel.ilie@upmc.fr

Djamel Eddine Saïdouni
MISC Laboratory, University Constantine 2, Ali Mendjeli Campus, 25000 Constantine, Algeria
e-mail: saidouni@misc-umc.org

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_42

based on an agent centric approach and a context-awareness. The proposed HoA model takes into account the major features and functionalities of AmI, in particular dynamic requirements: AmI systems can be open; agents can enter or leave the system.

This paper introduces an efficient planning management process into the architecture of the agent. In particular, we aim at offering to each AmI agent, a powerful predictive service, that can run on the fly. Like in other recent approaches, e.g. [6,9], which are dedicated to the planning and the validation of BDI MAS systems, we focus on one agent rather than on the whole MAS, since this eases us to embed agent in whatever environment and to deal with the openness of AmI systems.

We take profit from the fact that the plan of the agent can be derived from the current set of intentions, which result from the reasoning of the BDI interpreter. Our approach is based on a formal description language recently proposed for plans, namely *AgLOTOS* [3]. This allows us to introduce modularity and concurrency aspects to compose sub-plans. Unlike the formal description of [9], the *AgLOTOS* semantics overpasses the sequential execution of sub-plans. Rather, the concurrency of sub-plans is fully implemented, actually only being restrained to solve the possible inconsistencies of intentions.

In this paper, the semantics of *AgLOTOS* is enriched to automatically produce a *Contextual Planning System* (*CPS*), which allows to automatically guide the execution of plans. In contrast to [6], which restrains the execution to one possible subset of consistent intentions, the CPS is a state transition structure which captures whatever execution in respect to the predicted evolution of the context.

The paper is organized as follows: In Section 2, the *AgLOTOS* specification language is briefly described and used to associate plans with intentions (e.g. composition of plans). In Section 3, the *AgLOTOS* operational semantics is enriched to automatically produce the Contextual Planning System (*CPS*). A realistic scenario is given as an illustration of our guidance mechanism. The last section concludes and outlines our perspectives.

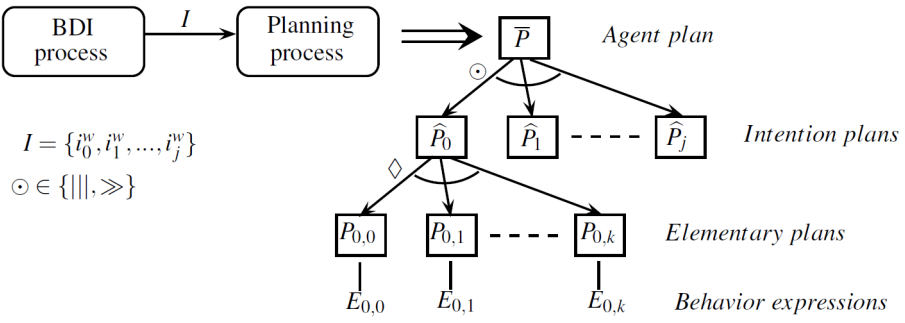


Fig. 1 Agent cycle

2 AgLOTOS Specification Language

The behavior of the BDI agent we consider in this paper is carried by two successive processes, as highlighted in Figure 1. As usual, the *BDI process* represents the reasoning mechanism, based on the beliefs (B), desires (D), and intentions (I) structures, the instances of which defines the BDI states of the agent. Triggered by the perceived events, the BDI process manages/updates the B,D and I structures. In order to organize its selected intentions, the BDI process is able to schedule them by associating each one a given weight (see Section 2.2).

From the set of intentions, the *Planning process* is called by the BDI process to produce a plan of actions, helped by a library of plans (*LibP*). In our approach for each BDI state, the plan of the agent, namely the *Agent plan* is composed in two levels: *(1)* The agent plan is made of sub-plans called *Intentions plans*, each one dedicated to achieve the associated selected intention ; *(2)* Each intention plan is an alternate of several sub-plans, called *Elementary plans*, extracted from the *LibP* library. This allows one to consider different ways to achieve the associated intention (see Section 2.1). Further, we assume that the *LibP* library is indexed by the set of all the possible intentions for the agent.

2.1 The Syntax of Elementary Plans

Elementary plans are written using the algebraic language *AgLOTOS* [2]. This language extends the LOTOS language [1] in order to deal with the concurrency of actions in plans.

Let $\mathcal{O}$ be the (finite) set of observable actions which are viewed as instantiated predicates, ranged over $a, b, ...$ and let L be any subset of $\mathcal{O}$. Let $\mathcal{H} \subset \mathcal{O}$ be the set of the so-called AmI primitives which represent the mobility and communication:

- In *AgLOTOS*, actions are refined to make the AmI primitives observable:*(1)* an agent can perceive the enter and leave of other agents in the AmI system, *(2)* it can move between the AmI system localities and *(3)* an agent can communicate with another agent in the system.
- An *AgLOTOS* expression refers to contextual information with respect to the (current) BDI state of the agent: *(1)* Θ is a finite set of space localities, *(2)* Λ is a set of agents with which it is possible to communicate, and $\mathcal{M}$ is the set of possible messages to be sent and received.
- The agent mobility is expressed by the primitive $move(\ell)$ which is used to handle the agent move to some locality ℓ ($\ell \in \Theta$). The syntax of the communication primitives is inspired from the semantics of the π-calculus primitives, however considered within a totally dynamic communication support, hence without specification of predefined channels: the expression $x!(\nu)$ specifies the emission to the agent x ($x \in \Lambda$) of some message ν ($\nu \in \mathcal{M}$), whereas, the expression $x?(\nu)$ means that ν is received from some agent x.

Let $Act = \mathcal{O} \cup \{\tau, \delta\}$, be the set of actions, where $\tau \notin \mathcal{O}$ is the internal action and $\delta \notin \mathcal{O}$ is a particular observable action which features the successful termination of a plan.

The *AgLOTOS* language specifies pairs for each elementary plan a name to identify it and an *AgLOTOS* expression to feature its behavior. Consider that elementary plan's names are ranged over $P, Q, ...$ and that the set of all possible behavior expressions is denoted $\mathcal{E}$, ranged over $E, F, ...$ The *AgLOTOS* expressions are written by composing (observable) actions through LOTOS operators. The syntax of an *AgLOTOS* elementary plan P is defined inductively as follows:

$$
\begin{array}{lll}
P ::= E & & Elementary\ plan \\
E ::= & exit \mid stop & \\
& \mid a; E \mid E \odot E & (a \in \mathcal{O}) \\
& \mid hide\ L\ in\ E & \\
\mathcal{H} ::= & \mid move(\ell) & (\mathcal{H} \subset \mathcal{O}, \ell \in \Theta) \\
& \mid x!(\nu) \mid x?(\nu) & (x \in \Lambda, \nu \in \mathcal{M}) \\
\odot & = \{ \mid\mid\mid, \mid[L]\mid, \mid\mid, [\,], \gg, [> \}
\end{array}
$$

The elementary expression $stop$ specifies a plan behavior without possible evolution and $exit$ represents the successful termination of some plan. In the syntax, the set $\odot$ represents the standard LOTOS operators: $E\,[\,]\,E$ specifies a non-deterministic choice, $hide\ L\ in\ E$ a hiding of the actions of L that appear in E, $E \gg E$ a sequential composition and $E\ [>\ E$ the interruption. The LOTOS parallel composition, denoted $E\,|[L]|\,E$, can model both synchronous composition, $E \,||\, E$ if $L = \mathcal{O}$, and asynchronous composition, $E \,|||\, E$ if $L = \emptyset$. In fact, the *AgLOTOS* language exhibits a rich expressivity such that the sequential executions of plans appears to be only a particular case.

2.2 Building the Agent Plans from Intentions and Elementary Plans

The building of an agent plan requires the specific *AgLOTOS* operators:

- at the agent plan level, the parallel $|||$ and the sequential $\gg$ composition operators are used to build, in respect with the intentions of the agent and the associated weights.
- the *alternate composition* operator, denoted $\Diamond$, allows to specify an alternation of elementary plans. In particular, an intention is satisfied iff at least one of the associated elementary plans is successfully terminated.

Let $\widehat{\mathcal{P}}$ be the set of names used to identify the possible intention plans: $\widehat{P} \in \widehat{\mathcal{P}}$ and let $\overline{\mathcal{P}}$ be the set of names qualifying the possible agent plans: $\overline{P} \in \overline{\mathcal{P}}$.

$$\widehat{P} ::= P \mid \widehat{P} \Diamond \widehat{P} \qquad \textit{Intention plan}$$
$$\overline{P} ::= \widehat{P} \mid \overline{P} \,|||\, \overline{P} \mid \overline{P} \gg \overline{P} \qquad \textit{Agent plan}$$

With respect to the set of intentions I of the agent, the agent plan is formed in two steps: *(1)* by an extraction mechanism of elementary plans from the library, *(2)* by using the composition functions called $options$ and $plan$:

- $options : \mathcal{I} \rightarrow \widehat{\mathcal{P}}$, yields for any $i \in \mathcal{I}$, an intention plan of the form: $\widehat{P}_i = \Diamond_{P \in libP(i)} P$.
- $plan : 2^I \rightarrow \overline{\mathcal{P}}$, creates the final agent plan $\overline{P}$ from the set of intentions I. Depending on how I is ordered, the intention plans yielded by the different mappings $\widehat{P}_i = options(i)$ $(i \in I)$ are composed by using the *AgLOTOS* composition operators $|||$ and $\gg$.

To be pragmatic considering any BDI state of the agent, we propose that the agent can label the different elements of the set I of intentions by using a weight function $weight : I \longrightarrow \mathbb{N}$. This allows us to weight the corresponding intention plans yielded by the mapping $options$. The ones having the same weight are composed by using the concurrent parallel operator $|||$. In contrast, the intention plans corresponding to distinct weights are ordered by using the sequential operator $\gg$. For instance, let $I = \{i_0^1, i_1^2, i_2^1, i_3^0\}$ be the considered set of intentions, such that the superscript information denotes a weight value, and let $\widehat{P_0}, \widehat{P_1}, \widehat{P_2}, \widehat{P_3}$ be their corresponding intention plans, the constructed agent plan could be viewed (at a plan name level) as: $plan(I) = \widehat{P_1} \gg (\widehat{P_0} ||| \widehat{P_2}) \gg \widehat{P_3}$.

A Simple AmI Example. Let us consider the *AmI Universitary scenario* presented in [2] where Alice and Bob are two agents. The proposed problem of Alice is that she cannot make the two following tasks in the same time: *(1)* to meet with Bob in the locality ℓ_1, and *(2)* to get her exam copies from the locality ℓ_2. Clearly, the Alice's desires are conflictual since Alice cannot be in two distinct localities simultaneously.

	Alice's scenario
	$I_A = \{meeting(Bob, \ell_1), asking(Bob, get\_copies(\ell_2))\}$
	$\overline{P_A} = meet(Bob); exit \;\|||\; Bob!(get\_copies(\ell_2)); exit$
	Bob's scenario
	$I_B = \{meeting(Alice, \ell_1), getting\_copies(\ell_2)\}$
	$\overline{P_B} = get\_copies(\ell_2); exit \;\gg\; move(\ell_1); meet(Alice); exit$

The scenarii of Alice and Bob are specified separately, assuming that Bob and Alice may coordinate in order to achieve their intentions, at their BDI process levels. The actions in plans are simply expressed by using instantiated predicates, like $get\_copies(\ell_2)$. Intention plans are composed from elementary plans which are viewed as concurrent processes, terminated by $exit$, *a la LOTOS*.

The BDI process can order the set of intentions to be considered. For instance, the intention set of Bob $I_B = \{meeting(Alice, \ell_1), getting\_copies(\ell_2)\}$ is ordered such that $weight(meeting(Alice, \ell_1)) < weight(getting\_copies(\ell_2))$. In the intention set I_B, the corresponding agent plan expression of Bob is: $\overline{P_B} = get\_copies(\ell_2); exit \gg move(\ell_1); meet(Alice); exit$, which is built by using the $options$ and $plan$ mappings. Pay attention that some actions can be processed concurrently, so is the case in the agent plan $\overline{P_B}$, for the two intention plans $get\_copies(\ell_2); exit$ and $move(\ell_1); meet(Alice); exit$.

3 Contextual Planning Management

3.1 Semantics of AgLOTOS

The *AgLOTOS* operational semantics is basically derived from the one of LOTOS. A pair (E, P) represents a process identified by P, such that its behavior expression is E. Basic LOTOS semantics is detailed in [2] which formalizes how a process can evolve under the execution of actions. In particular, the rule $\frac{P{:=}E \quad E \xrightarrow{a} E'}{P \xrightarrow{a} E'}$, specifies how an (E, P) pair is changed to (E', P) under any action a. Actually, $P := E$ means to consider any (E, P) source pair and $P \xrightarrow{a} E'$ means changing E to E' for P under the execution of a. As far as *AgLOTOS* is concerned, these rules also represent the operational semantics of elementary plans, viewed as processes.

The next definition specifies how the expression of an *agent plan* is formed compositionally from the expressions of the *intentions plans* of the agent, themselves built from an alternate of *elementary plans* and their behavior expressions. With respect to some agent plan $\overline{P}$, we introduce a notion of configuration of plans in order to specify that a part of the plan can already be executed. Further, the notation $[\overline{P}]$ represents the configuration of the agent plan $\overline{P}$, it is an *AgLOTOS* expression, which is obtained by composition of the different intention plan configurations of the agent, like $(E, \widehat{P})$.

Definition 1. Any plan configuration $[\overline{P}]$ has a generic representation defined by the following two rules:

1. $\dfrac{\overline{P}{::=}\widehat{P} \qquad \widehat{P}{::=}\Diamond^{k=1..n} P_k \qquad P_k{::=}E_k}{[\overline{P}]{::=}(\Diamond^{k=1..n} E_k, \widehat{P})}$
2. $\dfrac{\overline{P}{::=}\overline{P_1} \odot \overline{P_2} \qquad \odot \in \{|||, \gg\}}{[\overline{P}]{::=}[\overline{P_1}] \odot [\overline{P_2}]}$

The planning state of the agent is now defined contextually, taking into account the agent locality and a termination information about the different intention plans defined for the agent.

Definition 2. A (contextual) planning state is a tuple $(\mathcal{C}, \ell, T)$, where $\mathcal{C}$ is any plan configuration $[\overline{P}]$, ℓ corresponds to an expected locality for the agent, and T is the subset of intention plans which are terminated.

Table 1 Semantic rules of intention and agent configurations

Intention plan level		
(Action)	$\dfrac{E \xrightarrow{a} E' \quad a \in \mathcal{O} \cup \{\tau\}}{(E, \widehat{P}) \xrightarrow{a} (E', \widehat{P})}$	$\dfrac{E \xrightarrow{\delta} E'}{(E, \widehat{P}) \xrightarrow[\widehat{P}]{\tau} (E', \widehat{P})}$

Agent plan level		
(Action)	$\dfrac{\mathcal{C} \xrightarrow{a} \mathcal{C}' \quad a \in \mathcal{O} \cup \{\tau\}}{(\mathcal{C}, \ell, T) \xrightarrow{a} (\mathcal{C}', \ell, T)}$	$\dfrac{\mathcal{C} \xrightarrow[\widehat{P}]{\tau} \mathcal{C}'}{(\mathcal{C}, \ell, T) \xrightarrow{\tau} (\mathcal{C}', \ell, T \cup \{\widehat{P}\})}$
(Communication)	$\dfrac{\mathcal{C} \xrightarrow{x!(\nu)} \mathcal{C}' \quad x \in \Lambda}{(\mathcal{C}, \ell, T) \xrightarrow{x!(\nu)} (\mathcal{C}', \ell, T)}$	$\dfrac{\mathcal{C} \xrightarrow{x?(\nu)} \mathcal{C}' \quad x \in \Lambda}{(\mathcal{C}, \ell, T) \xrightarrow{x?(\nu)} (\mathcal{C}', \ell, T)}$
(Mobility)	$\dfrac{\mathcal{C} \xrightarrow{move(\ell')} \mathcal{C}' \quad \ell \neq \ell'}{(\mathcal{C}, \ell, T) \xrightarrow{move(\ell')} (\mathcal{C}', \ell', T)}$	$\dfrac{\mathcal{C} \xrightarrow{move(\ell)} \mathcal{C}'}{(\mathcal{C}, \ell, T) \xrightarrow{\tau} (\mathcal{C}', \ell, T)}$
(Sequence)	$\dfrac{\mathcal{C}_1 \xrightarrow{a} \mathcal{C}_1' \quad a \in \mathcal{O} \cup \{\tau\}}{\mathcal{C}_1 \gg \mathcal{C}_2 \xrightarrow{a} \mathcal{C}_1' \gg \mathcal{C}_2}$	$\dfrac{\mathcal{C}_1 \xrightarrow[\widehat{P}]{\tau} \mathcal{C}_1'}{\mathcal{C}_1 \gg \mathcal{C}_2 \xrightarrow[\widehat{P}]{\tau} \mathcal{C}_1' \gg \mathcal{C}_2}$
(Parallel)	$\dfrac{\mathcal{C}_1 \xrightarrow{a} \mathcal{C}_1' \quad a \in \mathcal{O} \cup \{\tau\}}{\mathcal{C}_1 \vert\vert\vert \mathcal{C}_2 \xrightarrow{a} \mathcal{C}_1' \vert\vert\vert \mathcal{C}_2}$	$\dfrac{\mathcal{C}_1 \xrightarrow[\widehat{P}]{\tau} \mathcal{C}_1'}{\mathcal{C}_1 \vert\vert\vert \mathcal{C}_2 \xrightarrow[\widehat{P}]{\tau} \mathcal{C}_1' \vert\vert\vert \mathcal{C}_2}$
	$\dfrac{\mathcal{C}_1 \xrightarrow{a} \mathcal{C}_1' \quad a \in \mathcal{O} \cup \{\tau\}}{\mathcal{C}_2 \vert\vert\vert \mathcal{C}_1 \xrightarrow{a} \mathcal{C}_2 \vert\vert\vert \mathcal{C}_1'}$	$\dfrac{\mathcal{C}_1 \xrightarrow[\widehat{P}]{\tau} \mathcal{C}_1'}{\mathcal{C}_2 \vert\vert\vert \mathcal{C}_1 \xrightarrow[\widehat{P}]{\tau} \mathcal{C}_2 \vert\vert\vert \mathcal{C}_1'}$

Table 1 shows the operational semantic rules defining the possible planning state changes for the agent. These rules are applied to produce a Contextual Planning transition System, called CPS, from an initial planning state, e.g. $([\overline{P}], \ell, \emptyset)$, meaning that the agent is initially at locality ℓ, and its plan configuration is $[\overline{P}]$. There are two kinds of transition rules:

Intention plan level: When an intention plan is assumed to be treated, the left hand side transition $(\mathcal{C}_1, a, \widehat{P}, \mathcal{C}_2)$, denoted $\mathcal{C}_1 \xrightarrow[\widehat{P}]{a} \mathcal{C}_2$, expresses a change of

intention configuration, from $\mathcal{C}_1$ to $\mathcal{C}_2$, and assumes the execution of the action a from $E \xrightarrow{a} E'$ and $P := E$. The right hand side transition highlights the termination case, keeping trace of the intention plan $\widehat{P}$ that is going to be terminated. By calling $\mathcal{CN}$ the set of all the possible intention plan configurations for the agent, the transition relation is a subset of $\mathcal{CN} \times Act \times \widehat{\mathcal{P}} \times \mathcal{CN}$. For sake of clarity, the transition $(\mathcal{C}_1, a, nil, \mathcal{C}_2)$ is simply denoted $\mathcal{C}_1 \xrightarrow{a} \mathcal{C}_2$. Observe that due to the fact we consider a predictive guidance in this paper, only successful executions are taken into account, thus abstracting that a plan may fail. Moreover, the semantics of the alternate operator is reduced to a simple non-deterministic choice of LOTOS: $\Diamond^{k=1..n} E_k \equiv [\,]^{k=1..n} E_k$ in order to possibly take into account every elementary plan in order to achieve the corresponding intention.

Agent plan level: the possible changes of the planning states, like $(\mathcal{C}, \ell, T)$, are expressed at this level. In the Communication rules, the action send $x!(\nu)$ (resp. receive $x?(\nu)$) is constrained by the visibility of the agent x in its neighborhood. In the Mobility rule, the effect of the $move(\ell')$ action yields the agent to be placed in ℓ'. The Action rules refer to the ones of the intention plan level. The left hand side one exhibits the case of a regular action, whereas the right hand side one specifies the termination case of some intention plan, which is added to T.

3.2 Planning Guidance

From any set of intentions in the agent, denoted I, a Contextual Planning System is built, by using the rules of Table 1 and taking into account contextual information of three kinds: *(1)* the reached locality in a planning state, *(2)* the set of intention plans that are terminated when reaching a planning state, and *(3)*, more globally, the set Λ of neighbors currently known by the agent.

Definition 3. The Contextual Planning System, denoted CPS, is a labeled kripke structure $\langle S, s_0, Tr, \mathcal{L}, \mathcal{T} \rangle$ where:

- S is the set of planning states,
- $s_0 = ([\overline{P}], \ell, \emptyset) \in S$ is the initial planning state of the agent, such that $[\overline{P}]$=$plan(I)$ and ℓ represents the current locality of the agent,
- $Tr \subseteq S \times Act \times S$ is the set of transitions. The transitions are denoted $s \xrightarrow{a} s'$ such that $s, s' \in S$ and $a \in \mathcal{O} \cup \{\tau\}$,
- $\mathcal{L} : S \rightarrow \Theta$ is the locality labeling function
- $\mathcal{T} : S \rightarrow 2^{\widehat{\mathcal{P}}}$ is the termination labeling function which captures the terminated intention plans.

In a CPS, the transitions from any state s only represent actions that are realizable. Like in STRIPS description language [6], actions to be executed are modeled by instantiated predicates submitted to preconditions and effects. In this paper, the preconditions only concern the contextual information known in that state. Let $pre(a)$

be the precondition of any action a, then $pre(x!(\nu)) = pre(x?(\nu)) = (x \in \Lambda)$ and for any other action a, $pre(a(\ell)) = \ell \in \mathcal{L}(s)$.

In order to guide the agent, the planning process can select an execution trace through the CPS which maximizes the number of intention terminations, with respect to the mapping $\mathfrak{T}$ in CPS states. This can be captured with the notion of Maximum trace, based on a trace mapping $end : \Sigma \longrightarrow 2^{\widehat{\mathcal{P}}}$ used to specify the set $end(\sigma)$ of the termination actions that occur in a trace $\sigma \in \Sigma$. From an algorithmical point of view, the configurations having the maximum number of terminated intention plans could be straightforwardly detected by parsing the CPS structure, with regards to the set of terminated intention plans of each built configuration. By labeling these configurations with a specific proposition `MAX`, the search of maximum traces is reduced to the traces which satisfies the (LTL) temporal logic property `AF(MAX)`.

The consistency of a set I of agent intentions can also be checked, in particular in two extreme cases:

- if $|end(\sigma)| = |I|$, means that all the intentions of I are consistent,
- if $|end(\sigma)| = 0$, there is no satisfied intention, so the agent plan $\overline{P}$ is contextually unappropriated with respect to the set of agent intentions.

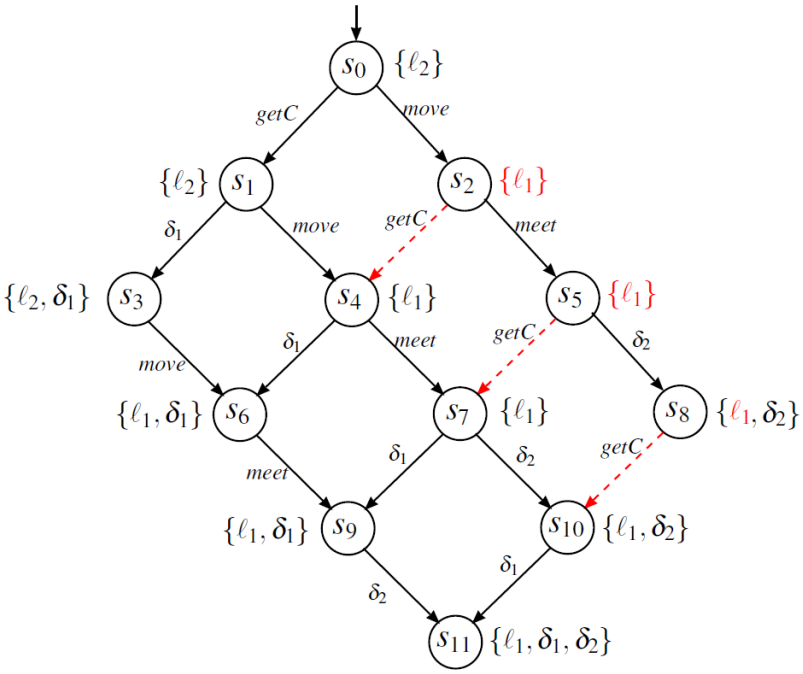


Fig. 2 The CPS_B corresponding to the plan $\overline{P_B}$

Application to the Scenario. We reconsider the scenario of Section 2 to achieve the intentions of Bob in a parallel way: $[\overline{P_B}] = ((E_g, \widehat{P_g})|||(E_m, \widehat{P_m}))$ is the agent plan configuration considered for Bob. The pairs $(E_m, \widehat{P_m})$ and $(E_g, \widehat{P_g})$ are two intention plan expression of Bob. The first one corresponds to the intention $meeting(Alice, \ell_1)$ and the second to $getting\_copies(\ell_2)$, such that $E_m = move(\ell_1); meet(Alice); exit$ and $E_g = get\_copies(\ell_2); exit$.

The Contextual Planning System of Bob, denoted CPS_B, is illustrated in Figure 2. It is built from the initial CPS state, $s_0 = ([\overline{P_B}], \ell_2, \emptyset)$, taking into account the current locality ℓ_2 of Bob. In the figure, the dashed edges represent the unrealized transitions from the states $s \in \{s_2, s_5, s_8\}$, because $pre(getC) = \ell_2 \notin \mathcal{L}(s)$.

An example of maximum trace derived from s_0 is the following, expressing that Bob got the copies before moving to the meeting with Alice:

$$((E_g, \widehat{P_g})|||(E_m, \widehat{P_m})) \xrightarrow{getc} ((E'_g, \widehat{P_g})|||(E_m, \widehat{P_m}), \ell_2, \emptyset) \xrightarrow[\widehat{P_g}]{\tau} ((E_m, \widehat{P_m}), \ell_2, \{P_g\})$$

$$\xrightarrow{move(\ell_1)}$$

$$((E'_m, \widehat{P_m}), \ell_1, \{P_g\}) \xrightarrow{meet} ((E''_m, \widehat{P_m}), \ell_1, \{P_g\}) \xrightarrow[\widehat{P_m}]{\tau} ((stop, \widehat{P_m}), \ell_1, \{P_g, P_m\})$$

4 Conclusion

The algebraic language AgLOTOS appears to be a powerful way to express an AmI agent plan as a set of concurrent processes, helped by an adapted plan library describing elementary plans. The main contribution of this paper is an enriched semantics of AgLOTOS: it allows to build a Contextual Planning System (CPS), for any BDI state of the agent. From the current set of intentions of the agent, all the possible plan evolutions can be evaluated through the CPS. This allows one to define an original predictive mechanism to contextually guide the agent in its future executions. In particular, we demonstrate how to realize the concurrent executions of the agent plans, while optimizing the number of satisfied intentions. Observe that the proposed techniques are also suitable for some class of partially ordered set of instances. Among the possible perspectives, we aim at combining the CPS approach with learning techniques like in [10], since this also can be viewed as a guidance mechanism but based on past-experiences.

References

1. Brinksma, E. (ed.): ISO 8807, LOTOS - A Formal Description Technique Based on the Temporal Ordering of Observational Behaviour (1988)
2. Chaouche, A.C., El Fallah Seghrouchni, A., Ilié, J.M., Saïdouni, D.E.: A higher-order agent model for ambient systems. Procedia Computer Science 21, 156–163 (2013)
3. Chaouche, A.C., El Fallah Seghrouchni, A., Ilié, J.M., Saïdouni, D.E.: A dynamical plan revisiong for ambient systems. Procedia Computer Science 32, 37–44 (2014)
4. Doyle, J.: Rationality and its roles in reasoning. Computational Intelligence 8, 376–409 (1992)

5. Guivarch, V., Camps, V., Péninou, A.: Context Awareness in Ambient Systems by an Adaptive Multi-Agent Approach. In: Paternò, F., de Ruyter, B., Markopoulos, P., Santoro, C., van Loenen, E., Luyten, K. (eds.) AmI 2012. LNCS, vol. 7683, pp. 129–144. Springer, Heidelberg (2012)
6. Meneguzzi, F., Zorzo, A.F., da Costa Móra, M., Luck, M.: Incorporating planning into BDI agents. Scalable Computing: Practice and Experience 8, 15–28 (2007)
7. Olaru, A., Florea, A.M., El Fallah Seghrouchni, A.: A context-aware multi-agent system as a middleware for ambient intelligence. MONET 18(3), 429–443 (2013)
8. Rao, A.S., Georgeff, M.P.: An abstract architecture for rational agents. In: Nebel, B., Rich, C., Swartout, W.R. (eds.) KR, pp. 439–449. Morgan Kaufmann (1992)
9. Sardina, S., de Silva, L., Padgham, L.: Hierarchical planning in bdi agent programming languages: a formal approach. In: AAMAS 2006, Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1001–1008. ACM, New York (2006)
10. Singh, D., Sardina, S., Padgham, L., James, G.: Integrating learning into a BDI agent for environments with changing dynamics. In: Toby Walsh, C.K., Sierra, C. (eds.) Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pp. 2525–2530. AAAI Press, Barcelona (2011)

Part XIII
WSRL'2014 Papers

A Parallel and a Distributed Implementation of the Core Paths Graph Algorithm

Domenico Pascarella, Salvatore Venticinque, Rocco Aversa, Massimiliano Mattei, and Luciano Blasi

Abstract. The problem of generating optimal flight trajectories in the presence of no-fly zones and/or obstacles is computationally expensive. It is usually solved off-line, at least for those parts which cannot satisfy real time constraints. Here we investigate the exploitation of two parallel programming techniques to reduce the lead time. The former employs some parallelization techniques for multi-core and/or multi-processor platforms. The latter is targeted to a distributed fleet of Unmanned Aerial Vehicles. Here the statement of the problem and preliminary development are discussed. A two-dimensional scenario is analysed by way of example to show the applicability and the effectiveness of the approaches.

Keywords: UAV, trajectory planning, Core Paths Graph, parallel algorithm, distributed algorithm.

1 Introduction

Generating optimal flight trajectories that are continuously differentiable and are compliant with the problem constraints is a common problem in unmanned aerial vehicles [1]. Some constraints derive from the specific mission scenario, such as the avoidance of no-fly zones or obstacles, other geometric constraints can be the turn radius, the climb and descent angles. These algorithms pose a not negligible computational cost which, while not affecting offline applications, has a remarkable impact on possible online applications. The online trajectory optimization involves

Domenico Pascarella
CIRA (Italian Aerospace Research Centre), Soft Computing Laboratory, Capua, Italy
e-mail: d.pascarella@cira.it

Salvatore Venticinque · Rocco Aversa · Massimiliano Mattei · Luciano Blasi
Second University of Naples, Department of Industrial and Information Engineering, Aversa, Italy
e-mail: {salvatore.venticinque,rocco.aversa,
massimiliano.mattei,luciano.blasi}@unina2.it

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_43

an online processing of the flight trajectories to allow for a time-variant mission scenario, such as the occurrence of new no-fly zones or obstacles. The problem is further complicated if a real time replanning to account any change into the mission objectives and scenarios have to be taken into account. Indeed, real time replanning represents a mandatory requirement in these kind of critical applications and implies the invariable completion of the reaction to an external stimulus (e.g., a change into the perceived situation) within an hard deadline. Thus, the reference problem for an online application is a real time planning and replanning problem, that is a dynamic optimization problem.

Current processors have more cores per micro-chip instead of single higher clock rates. Therefore, a promising method for the speed-up of the trajectory planning process is the parallelization of the computation. Moreover, such an algorithm may be fitted in with future developments related to collaborative networks of Unmanned Aerial Vehicles (UAVs). These are a suitable solution for complex planning in a distributed environment (i.e., partially controllable and not observable by a single entity) and with uncertain knowledge. As stated in [2], a loose collection of vehicles that have some objectives in common is a collaborative team. If the vehicles are working together to achieve a common objective, then they are a cooperative team. The main motivation for team cooperation stems from the possible synergy, as the group performance is expected to exceed the sum of the performance of the single UAVs. Such cooperation entails: global information, because each UAV can share its sensor information with the rest of the group via a communication network; resource management, because a cooperative decision algorithm allows efficient allocation of the overall group resources over the multiple targets; robustness, because the team can reconfigure its distributed architecture in order to minimize the performance degradation if a failure occurs. Here we investigate the parallelization of an algorithm that converts the trajectory optimization problem into a minimum cost path search in a weighted and oriented graph, called Core Paths Graph (CPG) [1]. The CPG defines a discrete set of admissible connection paths in the space domain. The weights of the CPG arcs are obtained solving convex quadratic programming optimization problems. The particular case of piecewise polynomial trajectories minimizing flight paths length is fully developed in [1].

The CPG process can exploit a distributed implementation that is endorsed on a fleet of cooperative UAVs. In detail, in this paper we are interested in extending the basic CPG algorithm discussed in [1] in order to reduce its lead time for online operations. We firstly propose a parallel variant that employs a single multi-core and/or multi-processor platform. In addition, we provide a preliminary analysis of a distributed strategy on a network of UAVs.

A few methods for the parallelization of trajectory planning have been discussed in the scientific literature. For example, [3] presents a parallelization approach of the AA* algorithm. Reference [4] introduces R*GPU, a parallel extension of R*, whereas a parallel extension of probabilistic roadmaps is reported in [5]. Several works are also available as regards the multi-vehicle routing problem by means of decentralized and cooperative management (e.g., [6, 7]). Anyway, this is the first proposal of a parallelized and a distributed extension of the CPG algorithm.

2 The Mathematical Problem Formulation

Let us consider a two-dimensional[1] space domain $\Delta \subseteq \mathbb{R}^2$ in which to fly, possibly non connected, with a boundary $\delta\Delta$ that is determined by the presence of no-fly zones and/or obstacles. A parametric trajectory function in Δ is denoted with

$$s(\cdot): \quad t \in [t_0, t_f] \rightarrow s(t) = (x(t), y(t), z(t)) \tag{1}$$

where (x, y, z) are the coordinates in a given Cartesian reference system. We denote with $\dot{s} = \frac{ds}{dt}$ the first-order temporal derivative of the position s and with $E = (s, \dot{s})$ the vector of positions and their first derivatives. A flight trajectory from $E_0 = (s_0, \dot{s}_0)$ to $E_f = (s_f, \dot{s}_f)$ in Δ is a continuously differentiable C^1 oriented curve θ_{E_0,E_f} in the independent variable t, connecting the point s_0 to the point s_f with assigned first derivatives (tangents) $\dot{s_0}$ and $\dot{s_f}$ and at the extremes. Thus, the flight trajectory θ_{E_0,E_f} has the following expression

$$\theta_{E_0,E_f} \triangleq \left\{ s_{[t_0,t_f]}(\cdot) \in C^1_{[t_0,t_f]}: \quad s(t_0) = s_0, \quad s(t_f) = s_f, \quad \dot{s}(t_0) = \dot{s}_0, \quad \dot{s}(t_f) = \dot{s}_f \right\} \tag{2}$$

Other essential properties for a flight trajectory (e.g., Δ-compatibility, Δ-admissibility, Σ-admissibility, controllability, etc.) are thoroughly described in [1].

The cost of every admissible flight trajectory θ_{E_0,E_f} is a real nonnegative function $C(\cdot)$, which may be related to the path length or height, to the fuel consumption, to the risk or to other variables inherent to the flight mission. The search of the optimal flight trajectory connecting E_0 to E_f consists in finding a flight trajectory $\theta^*_{E_0,E_f}$ that is compliant with the desired properties and is a solution of the minimization problem

$$\theta^*_{E_0,E_f} = \arg \min_{\theta_{E_0,E_f} \in \Theta^{\Delta,\Sigma}_{E_0,E_f}} C\left(\theta_{E_0,E_f}\right) \tag{3}$$

The set of all the Σ-admissible optimal flight trajectories connecting every Δ-compatible E with each other is called the Optimal Δ-Connection Set. The problem (3) and the search of the Optimal Δ-Connection Set are complex tasks because the searching space $\Theta^{\Delta,\Sigma}_{E_0,E_f}$ is infinite dimensional, the cost function can be nonlinear and non-convex and the constraints are generally nonlinear and non-convex.

Some assumptions and approximations are introduced in order to convert the optimization problem (3) into a minimum cost path search over a graph. A Δ-Core Paths Graph (ΔCPG) is a discrete approximation of the Optimal Δ-Connection Set, wherein the nodes are suitable Δ-compatible E vectors and the arcs are Σ-admissible optimal flight trajectories connecting the nodes. We denote with N_{CPG} the set of CPG nodes and with Υ_{CPG} the set of CPG arcs. The weight $W_{i,j}$ of the arc connecting the i-th node with the j-th node represents the cost of the optimal flight

[1] Here we will not consider a three-dimensional space domain, but the problem formulation can be directly extended to the three-dimensional case as reported in [1].

trajectory from E_i to E_j. Reference [1] proves that the infinite dimensional problem (3) to calculate the CPG weights $W_{i,j}$ can be converted into the following finite dimensional problem

$$W_{i,j} = \min_{\delta \in \Omega} \left(\theta^f_{E_i,E_j}(\delta) \right), \quad \forall E_i, E_j \in \mathrm{N}_{CPG} \tag{4}$$

wherein Ω is the set of parameters δ ensuring Σ-admissible flight trajectories.

Once a CPG has been built and the weights $W_{i,j}$ have been calculated, the optimal flight trajectory between two nodes of the graph can be determined adopting a search algorithm for a minimum cost path in a graph. In [1] polynomial functions are used for the flight trajectory parameterization. The trajectory $\theta^f_{E_i,E_j}(\delta)$ can be expressed as

$$\begin{aligned} \theta^f_{E_i,E_j}(\delta) \triangleq \{ & y(x) = \delta_1 x^{p-1} + \cdots + \delta_{p-1} x + \delta_p : \\ & x \in [x_0, x_f],\ \delta \in \Omega \subseteq \mathbb{R}^p,\ y(x_0) = x_0,\ y(x_f) = y_f, \\ & \dot{y}(x_0) = \dot{y}_0,\ \dot{y}(x_f) = \dot{y}_f \} \end{aligned} \tag{5}$$

The equalities constraints in (5) and the obstacle avoidance requirements can be converted in a matrix notation, as described in [1]. Moreover, if we assume that the cost of a trajectory is proportional to its length, the cost function can be represented as a quadratic function of δ and the optimization problem (4) can be converted into

$$\begin{aligned} W_{i,j} = & \min_{\delta \in \Omega \subseteq \mathbb{R}^p} \delta^T Q \delta \\ & \text{s.t.} \\ & A_I \delta \leq b_I \\ & A_I \delta \leq b_I \end{aligned} \tag{6}$$

The problem (6) is a convex quadratic programming problem and can be efficiently solved using quadratic programming algorithms. The matrices Q, A_I, A_E and the vectors b_I, b_E are accurately described in [1].

3 CPG Algorithm

The CPG computation includes first an offline processing phase, which entails the setting of the CPG topology and the evaluation of the weights of the arcs. The topology is built by choosing the set N_{CPG} of CPG nodes. A criterion can be a uniform grid of points and directions with increased density in the proximity of no-fly zones and obstacles. Once a cost function $C(\cdot)$ has been selected, the weights $W_{i,j}$ of the arcs can be determined by solving the problem (6) if we adopt polynomial trajectories.

The verification of the desired properties for $W_{i,j}$ is another important step. This verification is carried out into the following three different phases:

1. evaluation of the properties that can be directly verified on the relations between E_i and E_j (i.e., distance limitations among the extreme points s_i and s_j, difference limitations among the extreme directions $\dot{s}_i$ and $\dot{s}_j$, etc.): if these properties are not satisfied, there is not an arc between E_i and E_j;
2. evaluation of the solvability of the optimization problem (6): if this has no admissible solutions for (E_i, E_j), then there is not an arc between E_i and E_j;
3. evaluation of a posteriori checks on the optimal trajectory between E_i and E_j: if the solution of the problem (6) does not respect other Σ-properties (not verified at point 1, such as obstacles avoidance), then the optimal trajectory for the pair (E_i, E_j) is not feasible and there is not an arc between E_i and E_j.

The pseudo-code for the CPG algorithm is composed by Algorithm 7 and Algorithm 8. The parametric structure *scenario* contains the relevant environment data (i.e., the space domain, no-fly zones, obstacles, etc.), $\mathrm{N}_{CPG_{points}}$ðİŚČand $\mathrm{N}_{CPG_{directions}}$ðİŚŚare respectively the set of points and the set of directions of the CPG nodes and A_{CPG} is the adjacency matrix of the CPG. The check functions perform the properties verification.

The lead time of the CPG algorithm depends on the number of nodes $|\mathrm{N}_{CPG}| = \sum_{i=0}^{|\mathrm{N}_{CPG_{points}}|} |\mathrm{N}_{CPG_{directions}}|_i$, wherein $|\mathrm{N}_{CPG_{directions}}|_i$ is the number of directions for the i-th point. A theoretical estimation of the lead time should take into account the computational complexity of `CPG_arcs_computing` function. If we suppose that the number of directions is the same for every point, the computational complexity is

$$O\left(|\mathrm{N}_{CPG_{directions}}|^2 \cdot \left|\mathrm{N}_{CPG_{points}}\right| \cdot \left(\left|\mathrm{N}_{CPG_{points}}\right| - 1\right) \cdot T_{QP}\right) \qquad (7)$$

wherein T_{QP} is the computational complexity of the quadratic programming solver. For instance, primal-dual interior point methods for convex quadratic programming usually exhibit polynomial-time complexity with respect to problem dimension [8].

The online adaptation of the CPG algorithm solicits an enlargement of the CPG against any change in the scenario. The enlargement is in way of including the vertexes that are related to new no-fly zones or obstacles and the current vehicle position. As a consequence of the update of the CPG topology, `CPG_arcs_computing` function is performed in online fashion starting from the new nodes.

Algorithm 7. `CPG_algorithm` $(scenario)$

```
1 begin
2     (N_CPG_points, N_CPG_directions) ⟵ set_CPG_topology(scenario)
3     A_CPG ⟵ CPG_arcs_computing(N_CPG_points, N_CPG_directions)
```

Algorithm 8. CPG_arcs_computing $\left(\mathrm{N}_{CPG_{points}}, \mathrm{N}_{CPG_{directions}}\right)$

```
1  begin
2      for i ⟵ 1 to size(N_CPG_points) do
3          for j ⟵ 1 to size(N_CPG_directions) do
4              end_points ⟵ feasibility_points_check(i, j)
5              for k ⟵ 1 to size(end_points) do
6                  end_directions ⟵ feasibility_directions_check(end_points)
7                  for l ⟵ 1 to size(end_directions) do
8                      trajectory ⟵ quadratic_optimization(i, j, k, l)
9                      if trajectory ≠ null then
10                         admissibility ⟵
                           post-feasibility_check(trajectory, scenario)
11                         if admissibility = true then
12                             set_arc(i, j, k, l, cost(trajectory))
```

4 Parallel CPG Algorithm

The CPG algorithm is computationally heavy due to the size of the graph in a real scenario and some expedients are convenient in order to make the proposed approach less time-consuming, for an online adaptation of the algorithm. A proper choice of the discrete set of CPG nodes, of the Σ-properties and of the cost function can already reduce the required time for the CPG construction.

Anyway, the algorithm can benefit from the use of multi-core and/or multi-processor platforms since the CPG graph construction can be formulated as a parallel process. It does not present loop-carried dependencies, namely one iteration of the loop does not depend upon the results of other iterations of the loop and the loop can be executed in any order, even in parallel. There exists a strict one-to-one relation between the iteration order (i, j, k, l) and the evaluated arc, therefore a CPG parallel process does not imply any potential race condition. Indeed, the arc weights computations do not depend on the sequence or timing of the parallel threads.

An OpenMP implementation of a parallel CPG algorithm is addressed here. OpenMP is an Application Program Interface (API), jointly defined by a group of major computer hardware and vendors [9]. It provides a portable and scalable model for developers of shared memory and multi-threaded parallel applications. The API supports C/C++ and Fortran on several architectures.

The adopted programming language for the CPG algorithm is C, wherein the OpenMP consists of a set of compiler pragmas that control how the program parallelism works. The application begins with a single thread, that is the master thread. As the program executes, the application may encounter parallel regions (specified by the parallel directive) in which the master thread creates a thread team. Each thread member of the team executes within the parallel region. The end of the parallel region marks implicitly a barrier synchronization for all the threads, that will wait here until the last thread completes. Afterwards, the thread team is removed and the master thread continues the sequential execution.

The OpenMP for directive splits a for-loop into a parallel region so that every thread gets different sections of the loop and it executes its own sections in parallel. In the case of the CPG algorithm, one or more of the for-loops can be parallelized amongst a team of threads. Each thread can handle some iterations of the loop, i.e., some arcs of the graph. As a consequence of the one-to-one relations between iterations and arcs, the CPG data structure does not have to be protected from concurrent accesses and the overhead due to OpenMP parallelism management is minimum.

Here we report the parallel implementation of the second-level for-loop of the CPG algorithm by way of example. This loop scans all the directions starting from the current CPG point and proceeds with the feasibility checks and the arcs processing by executing the lower-levels for-loops. The OOQP (Object-Oriented software for Quadratic Programming) package [10] has been used as regards the resolution of the problem (6). OOQP is a C++ package for solving convex quadratic programming problems. It employs object-oriented programming techniques for a primal-dual interior point algorithm. It contains code that can be used âĂIJout of the boxâĂİ to solve a variety of quadratic programming problems. It can be used as a standalone off-the-shelf solver, as an embeddable solver (i.e., as an integrated code in a custom application, that interfaces with the OOQP solver by invoking dedicated subroutines) or as a development framework (i.e., to develop new quadratic programming solvers). Within this work, we have used OOQP as an embedded solver into a C application for CPG algorithm.

4.1 Test Scenario and Test Results

The considered test scenario is a two-dimensional example with two circular no-fly zones in a rectangular space region Δ (Fig. 1), so we suppose that the aerial vehicle moves only in a plane (i.e., at a constant height). We also assume that the CPG nodes $(s, \dot{s})$ are the combination of the 115 points marked with circles in (Fig. 1) and of a discrete set of 36 equally spaced directions identified with the star of arrows centred in the point $(-1, 3)$ in such a figure. This choice produces a set of 4140 nodes and a number Υ_{CPG} of arcs that is equal to $|\mathrm{N}_{CPG}| \cdot (\mathrm{N}_{CPG} - 1)$ in the case of a complete graph without auto-connections. Nevertheless, some nodes are not Δ-compatible and further reduction of the arcs is obtained by applying the following Σ properties: a distance limitation of 5 km among the connected points and a difference limitation of 70° among the extreme directions. Finally, we consider 8th-order polynomial trajectories to connect nodes, namely $p = 9$. Constraints on the radius of curvature of the vehicle have not been considered. (Fig. 1) shows two sample trajectories. They both start from the point $(-1, -1)$ with a direction of -40° and arrive at the point $(4, 0)$ (the former with a direction of -40°, the latter with a direction of 30°).

Table 1 reports the runtime performances for the proposed parallel CPG implementation. Measures have been collected in terms of the wall-clock time of the parallelized for-loop iteration. Measurements have been accomplished by means

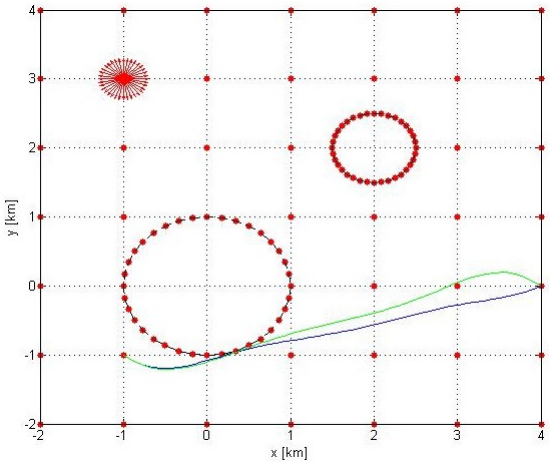


Fig. 1 Two-dimensional test scenario

of `omp_get_wtime` function. A general purpose workstation with Intel Core i5-2520M (two cores and a clock speed of 250 GHz) has been used as testbed. Each measure in Table 1 is the average of 10 runs. A single optimal trajectory is computed in about 300 ms. Best results have been obtained with 8 threads. The master thread is in charge of the remaining sequential part of the computation.

Table 1 Measures in seconds of the wall-clock time of the parallelized for-loop iteration

#Threads	Iteration #1	Iteration #2	Iteration #3	Iteration #4	Iteration #5
1	161.08	338.93	326.99	320.57	291.30
2	92.29	257.95	235.57	212.04	186.29
3	87.11	199.26	180.81	164.03	151.54
4	81.46	161.53	153.33	150.36	134.47
5	80.33	159.25	151.88	147.36	131.67
6	73.81	158.11	145.40	136.71	124.01
7	74.50	165.33	144.40	134.98	127.84
8	73.04	150.03	147.45	146.36	129.62

5 Distributed CPG Algorithm

A fleet of collaborative UAVs can be arranged as a distributed system, that is a system in which the individual UAVs are located on a network and communicate and

coordinate their actions by passing messages. In this case, the problem (6) can be distributed amongst the team of vehicles. Indeed, a distributed CPG algorithm would reduce the global processing time by taking advantage of the computing powers of the single vehicle platforms. Here we assume that the fleet is homogeneous, so the problem (6) has the same formulation and the same parameters for all the UAVs members.

The pseudo-code for the distributed strategy is composed by Algorithm 9 and Algorithm 8. In Algorithm 9, N_v is the number of vehicles in the team, $\mathrm{N}^{(i)}_{CPG_{points}}$ is the set of CPG points that are assigned to the i-th vehicle and $A^{(i)}_{CPG}$ is the adjacency matrix that is processed by the i-th vehicle. This strategy distributes the processing of a starting point of the trajectory. Afterwards, the single adjacency matrices are merged into the global matrix A_{CPG}.

Two policies have been considered for the allocation of CPG points: a geographic criterion and a fair criterion. The former assigns a CPG point to the closest UAV, which should be the most engaged vehicle by the processing of such a starting point. The latter randomly assigns CPG points in equal parts in order to supply a uniform distribution of the processing workload.

The distributed CPG algorithm has been simulated on a single platform by means of a variation of the parallel CPG algorithm. In more detail, a parallel OpenMP implementation of the first-level for-loop of `CPG_arcs_computing` function has been used in order to reproduce the distribution amongst the vehicles. Indeed, this loop scans all CPG points and processes the optimal flight trajectories starting from the current CPG point. Hence, a parallelization of the first-order for-loop of `CPG_arcs_computing` function is equivalent to the distribution of CPG points processing to some degree. The outcome is a double-level parallel CPG process.

Table 2 reports the test results with four vehicles. The first row concerns the sequential criterion. The test scenario and the testbed are the same of section 4.1. The vehicles are respectively located at the coordinates $(-1,-1)$, $(-1,3)$, $(3,-1)$ and $(3,3)$. The measurement of the global wall-clock time involves all the steps, from the setting of the topology to the merging of adjacency matrices.

Algorithm 9. `distributed_CPG_algorithm` $(scenario)$

```
1 begin
2     (N_CPG_points, N_CPG_directions) ⟵ set_CPG_topology(scenario)
3     (N^(i)_CPG_points) ⟵ assign_CPG_points_to_vehicles(N_v, N_CPG_points)
4     for i ⟵ 1 to size(v_i) do
5         A^(i)_CPG ⟵ CPG_arcs_computing(N^(i)_CPG_points, N_CPG_directions)
6     A_CPG ⟵ CPG_matrices_merge(A^(i)_CPG)
```

Table 2 Measures in seconds of the wall-clock time of the distributed CPG implementation

Criterion	Vehicle #1	Vehicle #2	Vehicle #3	Vehicle #4	Global
None	29708.98				29708.98
Geographic	18269.71	2410.31	1945.41	15492.43	18269.78
Fair	13276.62	13144.87	12591.35	13017.08	13276.71

6 Conclusion

Parallel and distributed extensions of a CPG algorithm have been proposed in order to improve the lead time for an online CPG application. Absolute performances of preliminary results do not allow for an on-line solution of the presented problem yet. However, we observed the scalability of the problem and the improvement obtained by a multi-threaded implementation over a shared memory architecture.

Further research will be conducted on the implementation on a target embedded device, eventually equipped with GPU accelerators. The design and development of a protocol for an effective CPG implementation on a fleet of UAVs is an additional objective to take into account the communication overhead in a distributed environment.

References

1. Mattei, M., Blasi, L.: Smooth flight trajectory planning in the presence of no-fly zones and obstacles. Journal of Guidance, Control, and Dynamics 33(2), 454–462 (2010)
2. Shima, T., Rasmussen, S.: UAV cooperative decision and control: challenges and practical approaches, Society for Industrial and Applied Mathematics, Philadelphia (2009)
3. Kopřiva, Š., Šišlák, D., Pěchouček, M.: Towards parallel real-time trajectory planning. In: Demazeau, Y., Müller, J.P., Rodríguez, J.M.C., Pérez, J.B. (eds.) Advances on PAAMS. AISC, vol. 155, pp. 99–108. Springer, Heidelberg (2012)
4. Kider, J.T., Henderson, M., Likhachev, M., Safonova, A.: High-dimensional planning on the GPU. In: Proceedings of 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 2515–2522 (May 2010)
5. Pan, J., Lauterbach, C., Manocha, D.: g-planner: real-time motion planning and global navigation using GPUs. In: Proceedings of AAAI Conference on Artificial Intelligence (2010)
6. Murray, R.M.: Resent research in cooperative control of multi-vehicle systems. Journal of Dynamic Systems, Measurement, and Control 129(5), 571–583 (2007)
7. Faied, M., Mostafa, A., Girard, A.: Vehicle routing problem instances: application to multi-UAV mission planning. In: Proceedings of AIAA Guidance, Navigation and Control Conference (August 2010)
8. Monteiro, R.D.C., Adler, I., Resende, M.G.C.: A polynomial-time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extension. Mathematics of Operations Research 15(2), 191–214 (1990)
9. OpenMP Application Program Interface, Version 4.0 (July 2013)
10. Gertz, E.M., Wright, S.J.: OOQP user guide (October 2001)

A Semantic Driven Approach for Requirements Verification

Gabriella Gigante, Francesco Gargiulo, and Massimo Ficco

Abstract. Requirements Engineering (RE) is a key discipline for the success of software projects. Consistency, completeness, and accuracy are the requirements quality properties to be guaranteed by the verification task in RE. An overview of the actual trends in RE is briefly summarized, focusing more closely on the requirements verification quality properties. Completeness results is the most difficult property to guarantee. It is hard to capture the software behavior against the whole external context. In the last years, research has focused its attention to the application of semantic Web techniques to the different tasks of RE. The adoption of ontologies seems promising to achieve the proper level of formalism and to argue on quality properties. This paper presents a survey of the main concepts that need to be accounted for requirement verification, and proposes an ontological engineering approach to demonstrate the overlapping of requirements against the external context.

Keywords: Requirements Engineering, Semantic Driven Approach, Ontologies.

1 Introduction

The emerging demands of actual research programs in ICT domain relate to the definition of new approaches to software development guaranteeing high level of quality with low costs. The success of software relies on its capability to implement what is needed and nothing else. The task of translating stakeholder needs into descriptions

Gabriella Gigante · Francesco Gargiulo
CIRA - Italian Aerospace Research Center Capua, Italy
e-mail: {g.gigante,f.gargiulo}@cira.it

Massimo Ficco
Department of Industrial and Information Engineering, Second University of Naples, via Roma 29, Aversa (CE), Italy
e-mail: massimo.ficco@unina2.it

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_44

of what software has to do is assigned to requirements [1]. As a consequence, requirement engineering plays a determining role in system and software engineering determining the project performance. The recent project RAMP studies the actual practices on requirement engineering in industry focusing on verification. It highlights that requirements are written in the most cases in natural language. Reviews are the most commonly used means to verify requirements, and the review process is expensive and effective only with the intervention of domain experts. Teams still face difficulties in the transition from theory to practice: formalization of requirements, consistency of textual requirements, requirements that describe solutions, definition of specification models [3]. Errors in requirements typically make up 25% to 70% of total software errors. The two thirds are detected after delivery. Their fixing cost can be more over than a third of the total production cost. Their ambiguity produces waste of time and repeated work. Their incompleteness usually leads to project failure making impossible the project secure planning and its monitoring [17] and to system failure [42]. Their insufficient levels of abstraction constrains project to premature choice of implementation details.

The idea discussed in this paper leverages on the adoption of ontologies to verify the requirement completeness against what can be considered the 'external context' of the specific software. It starts from the recent studies on semantic web techniques and intends to investigate if concepts related to semantic distance between RDF (Resource Description Framework) triplets can allow any evaluation of completeness. The advantage of representing the external context of software by means of ontologies is twofold: it can allow arguing on completeness in a quite formal way, and it can allow reuse. Besides it can provide a compositional approach to completeness itself providing verification against 'blocks of external context', evaluating when it can be 'sufficient' for the application domain.

2 Background

2.1 Requirement Engineering

The term Requirements Engineering (RE) has been probably introduced in 1979 in a technical report on software engineering for USA defense [4]. It has become a research topic since the 1990s. Requirement Engineering is facing many research concerns: scaling, security, safety, self-adaptation, strong coupling with software environment, tolerance and the persistent methodological problem [2]. The definition of Zave [5] gives rise to largely agreed distinct activities within RE: elicitating, modeling and analysis, verifying and validating, managing requirements. Verification activity demonstrates the compliance of requirements against the input. In this paper, we prefer allocate to it the check of ambiguity, consistency and completeness of requirements, as well as it aims to verify if the analysis and consequently the identified requirements are well-formed on the basis of the input and the applicable standards and guidelines.

2.2 Quality of Requirements

Correctness means that system implements the real needs of the users. If system really implements requirements, and requirements really describe customer needs, system really meets user needs. Thus requirements correctness assure the system correctness. The term correct itself is vague. Literature converges to define correctness as the combination of consistency and completeness [9]. Consistency requires that two or more requirements do not contradict each other: a requirement statement can be a direct or indirect refutation of more than one previous requirement. Such consideration suggest that consistency cannot be a binary relation and its identification is a semantic task. Looking at consistency as a more-wide ranging problem a general definition is pointed out by literature: "*to denote any situation in which a set of descriptions does not obey some relationship that should hold between them*" [10]. Relationship becomes consistency rules against which the descriptions can be checked. Relations could be defined between different knowledge domains, between different artifacts of a development process, between different statements within a single artifacts. [9] presents an interesting overview of consistency concerns related to the diagnosis task. In particular, research focuses on four main concerns [14]: *the detection of overlap* - if there is no overlapping, elements cannot be inconsistent; overlapping is detected by four approaches: representation conventions, shared ontologies, human inspection, and forms of similarity analysis; *the detection of inconsistencies* - rules can be checked according to a logic-based approach, to the model checking approach, to the specialized model analysis approach, and to the human-centered collaborative exploration approach; *the diagnosis of inconsistency* - it identifies the source, the cause and the impact; and *the handling of inconsistencies*. Recent trends push the adoption of domain ontologies to detect inconsistencies. Abduction is proposed as formal reasoning techniques for detecting and repairing inconsistencies [12]. [13] and [11] consider that up to now any RE tool makes use of such approaches. Completeness requires that a specification entails everything that is known to be 'true' in a certain context. Davis states that completeness is the most difficult of the specification attributes to define and incompleteness of specification is the most difficult violation to detect. Boehm lists the three fundamental characteristics of a complete document: (1) no information is left unstated or to be determined, (2) the information does not contain any undefined objects or entities, (3) no information is missing from this document [15]. The first two properties are referred to as internal completeness. The third property concerns the external completeness of the document. External completeness implies that specification addresses all of the information required for problem definition. At the beginnings formality is addressed as a solution to the problem. But Leveson [43] argues any specification language cannot lead alone to completeness of specification depending very much on of the formal specification language itself. Propositional logic notation does not scale well to complex expressions in terms of readability and tabular representation (AND/OR tables) can be reductive. Literature agrees that "*the only truly complete specification of something would be the thing itself*". The concept of sufficient completeness can be an efficient compromise, where sufficient depends on the type of system

being implemented. For example, standards related to safety critical software stress the check of external completeness. ECSS (European Cooperation on Space Standardization) [41] requires the identification of environmental conditions, operational scenarios, the definition of FDIR strategy and the allocation of all requirements to the next step. DO-178C [6] requires that software implements exactly what is needed and does not implement what is not required. It requires compliance to external environment by means of compliance to system requirements and to robustness quality and to safety requirements. Cesar project [16] proposes an overall vision of requirements completeness assuring that all viewpoints are addressed. It considers two sources: the system and constraints coming up from internal software team (Fig. 1).

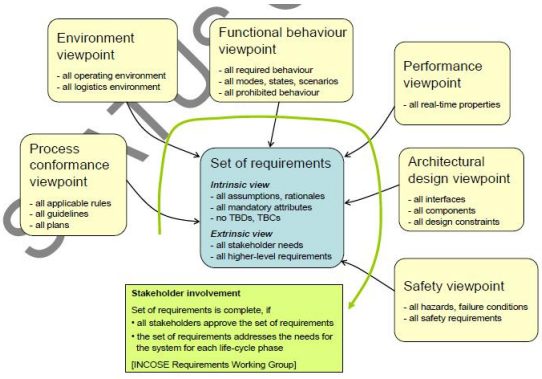


Fig. 1 Schema of requirements completeness according to Cesar project

Although some approaches have been proposed to achieve completeness up to now, no way to determine requirements completeness has been defined. Another problems related to RE is the test of completeness [13]. Ambiguity is also a quality attribute that cannot be neglected when defining requirements in natural language. There are two basic interpretations of ambiguity: the capability of being understood in two or more possible senses or ways; and the uncertainty [18]. Research on ambiguity goes hand in hand with Word Sense Disambiguation (WSD). The direct application of WSD techniques has led to the development of many usable tools in RE. Tools like QuaARS (Quality Analyzer of Requirement Specification) and ARM (Automated Requirement Measurement), developed by NASA (National Aeronautics and Space Administration) identify text quality indicators, such as weak and ambiguous phrases in requirement texts. Other approaches to ambiguity measures the indices of ambiguity of words or of sentences in a text, basing on the number of semantic meanings of words [18].

2.3 Ontologies

Ontologies are defined as an explicit formalized shared specification of a conceptualization. Conceptualization is an abstract interpretation of a domain of interest. The adjective 'formalized', means that it is machine readable, and 'shared', means that it is agreed by domain experts [7, 39]. Research streams has been relating to different concerns: the definition of ontologies elements, the definition of methodologies and tool following the overall ontology lifecycle (development, management, validation and integration), the building of specific ontologies to different purposes, the definition of languages to express ontologies. Guarino proposes a classification according to the generality level [7]: *highâĂŞlevel (or Upper-level) ontologies*: describe general concepts (space, time, material) independently of a specific domain or problem; *domain ontologies*: describe the vocabulary related to a generic domain; *task ontologies*: describe the vocabulary related to a generic task of the domain; and *application ontologies*: describe concepts belonging simultaneously to a domain and a task. Ontology engineering (or ontology building) is a subfield of knowledge engineering that investigates the methods for developing ontologies. Ontologies are developed mainly according to two approaches: the experience-based and the engineered-based. Ontology engineering offers directions towards the maintenance that is fundamentally difficult, the integration and the interoperability of ontologies [17]. In order to make ontologies machine readable different languages exist, usually are declarative languages. Languages basing on first order logic have much expressive power, but problems with decidability. Before OWL, much research has been conducted to define powerful ontology modeling languages. Research has begun with the XML-based RDF and RDF/S, progressed to the Ontology Inference Layer (OIL) and continued with the creation of DAML+OIL, joining the American proposal DAML-ONT5 with the European language OIL. RDF adds metadata on the resources on the Web. RDF is capable of representing data and exchanging knowledge over the Web in a way that is accessible to machines. It is recommended by the W3C, and it uses URIs to identify resources or things (the root of an ontology is called a thing). The RDF models knowledge by means of triplets $< subject, predicate, object >$ where subject is a resource the object is the value of resource and the predicate is the property between resource and its value. It is possible to say that $< subject >$ has a property $< predicate >$ valued by $< object >$. In an RDF graph, all triplets 'nodes and arcs' should be labeled with qualified URIs. OWL (Ontology Web Language) is a language for processing Web information, became a W3C recommendation in February 2004 and has been built using RDF to remedy weaknesses in RDF/S and DAML+OIL. W3C classifies OWL into three sublanguages, each of which is intended to supply different aspects of some incompatibilities: OWL Lite, OWL DL and OWL Full [8].

2.4 Ontologies in Engineering and in Requirement Engineering

The use of ontologies to provide a single and shared representation of knowledge for all system components has been largely motivated in literature in the last decade. An interesting review of the state of art of ontology-based software engineering can be found in [7, 17, 19], and the last recent proceedings of international forums like SWESE, W3C, SEKE discussing synergies between ontology engineering and software engineering, in particular between ontologies and requirement engineering. There are a number of benefits to use ontologies in RE. Literature proposes at least three ways of using ontologies in RE. [11] and [17] provides an overview of ontologies in RE. Ontologies can be used to represent the application domain knowledge on which requirements specification should be built. In this sense, domain ontologies support elicitation, modeling and analysis allowing to reason on consistency, completeness, redundancy, and ambiguity. In the railway domain, ontologies are pushed in the last years to provide train systems interoperability [23], to perform fault classifications on railway vehicles [21], as proposes in the project 'InteGRail' [20], to model requirements in natural language as (European Rail Traffic Management System-ERTMS) [22, 41]. In the automotive domain the idea of 'Automotive Ontology' at the core of the carâĂŹs information system is pervasive since 2008. Ontologies are developed to support on-board context awareness system defining individual observations from sensor data [24]. In the aerospace industry domain ontologies are a constant in each approach, but they are rarely defined. An important work describing a basic ontology for aerospace is presented in [28], where the basic concepts of functions, entities and problems are defined. Specific ontologies are proposed to support the justification of design [29], the aerospace composite manufacturing domain [30]; to define UAV missions [31] and to support the intelligence, surveillance, and reconnaissance (ISR) mission; to share knowledge of different units within in a company [25]; to support software development in Air Traffic Management. NASA addresses the use of ontologies in different contexts. The CDXA program aims to integrate knowledge in complex programs proposing a constellation of ontologies defined to overcome NASA data integration problems [26, 27]. The EU CESAR project develops a tool supporting requirements elicitation, specification and verification joining the boilerplates approach to domain ontologies [33]. NextGen project develops a tool for requirement engineering by defining requirements as wiki pages annotated by RDF triple to provide easy traceability and information retrieval. The ONTOREM project involving Airbus defines an ontology-driven Requirements Engineering Methodology (methods, processes and tools) recently applied to aircraft operability. Finally, it is relevant the initial work to apply ontologies to safety analysis like FMEA, and the challenging possibility to adopt safety ontologies to bridge the gap between the safety analysis and the system specification [32].

3 Completeness Verification

The proposed approach intends to evaluate completeness of a set of requirements against the 'external context' by using ontologies, being the external context a set of requirements too. The approach is essentially based on the intuition of modeling each textual requirement as set of RDF triplets. By means the RDF triplet extraction we move from an unstructured textual representation to a structured RDF representation of each requirement. Of course, the two representation are not semantically equivalent in strict sense, but the key concepts of requirement and the relationships among them are captured. Finally, both the requirements document and the external context will be represented by means of a set of RDF triplets sets. Since, the approach shifts the verification of completeness to the analysis of a set of requirements from the textual representation to the RDF representation, making the implicit assumption that the latter representation is 'enough expressive' with respect to the analysis that will be performed. Of course, the main advantage of this approach is the possibility to design a strategy for an 'automatic analysis' of the requirements. On the other hand, we have to explain the meaning of 'enough expressive' and we have to demonstrate the effectiveness of the approach. Let us give some preliminary definitions about our idea of 'semantic source' and 'semantic atoms'. Given a requirement/document: $R = \{s_1, \hat{a}\breve{A}\c{e}, s_n\}$, where s_i is a sentence of R, the first step is to represent each s_1 as: $s_i = \{t_1, \hat{a}\breve{A}\c{e}, t_k\}$, where t_j is a RDF triplet $[subj, pred, obj]$. Here, *subj*, *pred* and *obj* are respectively the subject, the predicate and the object of the sentence. In this model, we thus propose to use RDF triplet to describe a semantic related to the sentence S_i. Finally, the requirement/document R will be represented as a superset of set of RDF triplets. The extraction of the RDF triplet from a sentence is not an easy matter likewise all problems related to the elaboration of natural language and number of heuristics for this task are proposed. In addition, before the extraction, some other well-known NLP (Natural Language Process) tasks would be executed, such as: lemmatization, NER (Named Entity Recognition) both for common entities (i.e., people, places and organization) and domain specific entities; compound words, abbreviations or acronyms detection; the word sense disambiguation, etc. These tasks often rely on one or more general purpose or domain specific ontologies. Therefore, the *subj*, *pred* and *obj* are not plain words, but they represent concepts of an ontology. After the RDF representation of the requirements the second basic step is the definition of a similarity measure among RDF triplets. If we assume that *subj*, *pred* and *obj* belong to an ontology, a natural choice could be the adoption of well-known semantic metrics such as: Wu and Palmer [38], Leacock and Chodorow [35], Resnik [37], Lin [36], Jiang and Conrath [34]. The smaller the distance between two concept the more similar the concept are. Other domain specific metrics can be defined and the proposed approach is independent with respect the metric adopted. Consequently, it is possible to define a semantic distance between two RDF triplets, in which:

$$T_1 = [subj_1, pred_1, obj_1] T_2 = [subj_2, pred_2, obj_2], \quad (1)$$

using the distances between the concept contained in the triplets. A family of suitable distances is:

$$D(T_1, T_2) = \alpha d_1(subj_1, subj_1) + \beta d_2(pred_1, pred_2) + \gamma d_3(obj_1, obj_2), \quad (2)$$

where $\alpha, \beta, \gamma \in R$ and $\alpha + \beta + \gamma = 1$, d_i are distances between concept mentioned above, and their linear combination D is a metric. Then, given T as the set of all RDF triplets (T, h) is a metric space. In a specific context, the choice of the values of the coefficients α, β, γ allows to assign the preferred weight to the subject, the predicate and the object of a triplet and the choice of the d_i enables the use of the different distances.

In our approach, software high level requirements (HLR) shall be verified against system requirements (SR) allocated to software representing the external context. Let us give an example. The requirement "*the on-board software shall decode the telecommands according to CCSDS protocol within 10 ms in the flight phase*" can be translated into the following triplets: $T_1 = [FSW, decode, TC], T_2 = [TC, comply, TCStd]$, $T_3 = [T_1, last, max10ms]$, $T_4 = [T_3, execute, FLIGHT]$. A similarly transformation is applied to a system requirement. By means of the evaluation of distances between the set of triplets corresponding to HLR and SR, considerations on the HLR themselves shall be done. For instance, very variables distances measures can mean dis-homogenous HLR respect to the level of detail; distances equal to 0 can imply a total redundancy of the HLR against the SR, that is, it does not add any major detail; distances under a certain threshold can imply a semantic overlap, meaning a proper completeness to satisfy the SR.

4 Conclusions and Future Works

Requirement engineering is considered a key discipline for the success of software projects. Current practices in this discipline still reveal a difficult to implement the research approaches. Completeness is the most difficult property to guarantee. Semantic Web techniques seems to be very promising to bridge such gap. Ontologies are adopted to verify the consistency of requirements and support analysis for completeness. This paper presents a survey of the main concepts needed to evaluate the completeness of requirements against the 'external word' represented by 'blocks of knowledge'.

In the future work, we will try to evaluate completeness of a set of software requirements (HLR) of an on board software for a space mission, against a set of system requirements (SR), representing the 'nearest' external context. Such approach will be applied in different steps. The first step will try to evaluate thresholds applying the analysis to a set of HLR assessed for completeness against the related SR. Then, we will use the measured thresholds to evaluate completeness of not reviewed HLR. Thirdly, the set of triplets representing the external context will be enriched of the specific views envisaged in Fig. 1 the on-board domain block of knowledge representing the Pus standard [40], and the system safety analysis. In this way, we will guarantee a 'sufficient' completeness.

Acknowledgements. This work has been partially supported by the MIUR in the framework of "Potenziamento di laboratori pubblico-privato", PON Ricerca e CompetitivitÃă 2007-2013 (DISPLAY project) and the European CommunityâĂŹs Seventh Framework Programme as part of the ICT CoSSMic project (FP7-ICT-608806) and the CRYSTAL project (Critical System Engineering Acceleration), funded from the ARTEMIS Joint Undertaking under grant agreement no. 332830.

References

1. SWEBOK Guide to the Software Engineering Body of Knowledge. IEEE Computer Society (2004)
2. Cheng, B., Atlee, J.: Research Directions in Requirement Engineering. In: FOSE 2007, Future of Software Engineering, pp. 285–303 (2007)
3. Fanmuy, G., Fraga, A., Llorens, J.: Requirements verification in the industry. In: Hammami, O., Krob, D., Voirin, J.-L. (eds.) Complex Systems Design & Management, vol. 91, pp. 145–160. Springer, Heidelberg (2012)
4. Alfor, M., Lawson, J.: Software Requirements Engineering Methodology (Development). TRW Defense and Space Systems Group (1979)
5. Zave, P.: Classification of Research Efforts in Requirements Engineering. ACM Computing Surveys (CSUR) 29(4), 315–321 (1997)
6. RTCA, DO-178C. Software Consideration. In: Airborne Systems And Equipment Certification, Washington (December 2011)
7. Calero, C., Ruiz, F., Piattini, M.: Ontologies in Software Engineering and Software Technology. Springer (2005)
8. Taye, M.M.: Web-Based Ontology Languages and its Based Description Logics. The Research Bulletin of Jordan ACM II(II), 1–9
9. Zowghi, D., Gervasi, V.: On the Interplay Between Consistency, Completeness, and Correctness. Requirements Evolution, Journal of Information and Software Technology 45 (2003)
10. Nuseibeh, B., Easterbrook, S., Russo, A.: Leveraging Inconsistency in Software Development. Software Development Computer 33(4), 1–33 (2000)
11. Sharma, S., Pandey, S.: Integrating AI techniques in Requirement Phase: A Literature Review. In: IJCA Proceedings on 4th International IT Summit Confluence 2013 - The Next Generation Information Technology Summit Confluence, pp. 21–25 (2013)
12. Zhu, A., Jin, A.: Inconsistency Measurement of Software Requirements Specifications: An Ontology-Based Approach. In: Engineering of Complex Computer Systems, pp. 402–410 (2005)
13. Siegemund, K., Thomas, E., Zhao, Y., Pan, J., Assmann, U.: Towards ontology-driven requirements engineering. In: Workshop Semantic Web Enabled Software Engineering at 10th International Semantic Web Conference (ISWC), pp. 1–6 (2011)
14. Spanoudakis, G., Zisman, A.: Inconsistency Management in Software Engineering: Survey and Open Research Issues. In: Handbook of Software Engineering and Knowledge Engineering, pp. 329–380 (2001)
15. Boehm, B.W.: Verifying and validating software requirements and design specifications. IEEE Software (1), 75–88 (1984)
16. CESAR_D_SP2_R3.3_M3_Vol4_v1.000_PU Project, `http://www.cesarproject.eu/`
17. Castaneda, V., Ballejos, L., Caliusco, M., Galli, M.: The Use of Ontologies in Requirements Engineering. Global Journal of Researches in Engineering 10 (6) (Ver 1.0), 2–7 (2010)
18. Ceccato, M.: Ambiguity Identification and Measurements in Natural Language Texts
19. Gasevic, D., Kaviani, N., Milanovi, M.: Ontologies and Software Engineering. In: International Handbooks on Information Systems, pp. 593–615. Springer (2009)
20. Shingler, R., Fadin, G., Umiliacchi, G.P.: From rcm to predictive maintenance: The integrail approach. In: 4th IET International Conference on Railway Condition Monitoring, pp. 1–5 (2008)

21. De Ambrosi, C., Ghersi, C., Tacchella, A.: An ontology-based condition analyzer for fault classification on railway vehicles. In: Chien, B.-C., Hong, T.-P., Chen, S.-M., Ali, M. (eds.) IEA/AIE 2009. LNCS, vol. 5579, pp. 449–458. Springer, Heidelberg (2009)
22. Lodemann, M., Luttenberger, N.: Ontology-Based Railway Infrastructure Verification. In: Proceeding KMIS 2010, pp. 176–181 (2010)
23. Verstichel, S., Ongenaea, F., Loeve, L., Vermeulen, F., Dings, P., Dhoedt, B., Dhaene, T., De Turck, F.: Efficient data integration in the railway domain through an ontology-based methodology. Transportation Research Part C: Emerging Technologies 19(4), 617–643 (2011)
24. Kannan, S., Thangavelu, A., Kalivaradhan, R.: An intelligent driver assistance system (idas) for vehicle safety modelling using ontology approach. International Journal of Ubicomp (2010)
25. Lanfranchi, V., Bhagdev, R., Chapman, S., Ciravegna, F., Petrelli, D.: Extracting and Searching Knowledge for the Aerospace Industry. In: ESTC (2007)
26. Bonasso, R., Boddy, M., Kortenkamp, D., Bell, S.: Ontological Models To Support Space Operations
27. Keller, R., Berrios, D., Wolfe, S., Hall, D., Sturken, I.: Semantic Integration of Heterogeneous NASA Mission Data Sources
28. Malin, J., Throop, D.: Basic Concepts and Distinctions for an Aerospace Ontology of Functions, Entities and Problems. In: Aerospace Conference. IEEE (2007)
29. Kuofie, E.J.: RaDEX: A Rationale-based Ontology for Aerospace Design Explanation. Master of Science Programme Business Information Technology University of Twente
30. Verhagen, W., Curran, R.: Ontological Modelling of the Aerospace Composite Manufacturing Domain in Improving Complex Systems Today, pp. 215–222 (2011)
31. Schumann, B., Scanlany, J., Fangohrz, H.: A Generic Unifying Ontology for Civil Unmanned Aerial. In: 12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSM (2012)
32. Dittmann, L., Rademacher, T., Zelewski, S.: Performing FMEA Using Ontologies. In: 18th International Workshop on Qualitative Reasoning, Evanston, USA, pp. 209–216 (2004)
33. Bogusch, R., Gerlach, S.: Optimierungen in Requirements-Engineering in der Praxis
34. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy (1997)
35. Leacock, C., Chodorow, M.: Combining local context and WordNet similarity for word sense identification. WordNet: An Electronic Lexical Database 49(2), 265–283 (1998)
36. Lin, D.: An information-theoretic definition of similarity. In: Proceedings of the 15th International Conference on Machine Learning, vol. 1, pp. 296–304 (1998)
37. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy (1995)
38. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics, pp. 133–138 (1998)
39. Gruber, T.R.: A translation approach to portable ontologies. Knowledge Acquisition 5(2), 199–220 (1993)
40. Zazzaro, G., Gigante, G., Zaccariello, E., Ficco, M., Di Martino, B.: Supporting Development of Certified Aeronautical Components by applying Text Analysis Technique. In: Proceedings of the International Conference on Complex, Intelligent and Software Intensive Systems, CISIS-2014 (July 2014)
41. Venticinque, A., Mazzocca, N., Venticinque, S., Ficco, M.: Semantic support for log analysis of Safety-Critical embedded systems. In: Proc. of the 13th European Dependable Computing Conference (EDCC 2014), Newcastle, UK (May 2014)
42. Ficco, M., Daidone, A., Coppolino, L., Romano, L., Bondavalli, A.: An event correlation approach for fault diagnosis in SCADA infrastructures. In: Proc. of the 13th European Workshop on Dependable Computing (EWDC 2011), pp. 15–20 (2011)
43. Leveson, N.: Completeness in formal specification language design for process-control systems. In: Proceedings of the Third Workshop on Formal Methods in Software Practice, pp. 75–87 (2000)

A View-Based Acces Control Model for EHR Systems

Mario Sicuranza, Angelo Esposito, and Mario Ciampi

Abstract. Electronic Health Record (EHR) systems have the aim to collect clinical documents about patients, which typically contain very sensitive information. In order to manage who can do what on such clinical documents in the system, it is necessary to use a security mechanism. The Access Control (AC) goal is to guarantee the confidentiality and integrity of the data, and to allow the definition of security policies which reflect the need for privacy. In this paper, we define an innovative access control model that allows, on one hand, to meet the main requirements for EHR systems, and on the other hand to permit patients to define in detailed and clear manner the privacy policies on their clinical documents. The main innovation of this work is the principle of least privilege philosophy usage in the information content of the clinical documents. This feature allows to define an access control model that increases the patients' trust in the EHR system.

Keywords: Electronic Health Record, EHR-S, Access Control Model, View Based, Principle of least privilege.

1 Introduction

An Electronic Health Record system (EHR-S) is aimed to the collect and distribute electronic clinical documents and data about an individual's lifetime health status

Mario Sicuranza · Angelo Esposito
University of Naples "Parthenope", Department of Engineering, Naples, Italy,
and Institute of High Performance Computing and Networking,
National Research Council of Italy, ICAR-CNR, Naples, Italy
e-mail: {mario.sicuranza,angelo.esposito}@na.icar.cnr.it

Mario Ciampi
Institute of High Performance Computing and Networking, National Research Council of Italy,
ICAR-CNR, Naples, Italy
e-mail: mario.ciampi@na.icar.cnr.it

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_46

[1]. The EHR-S manages sensitive data, that have to be protected from unauthorized access. For this reason, the confidentiality of data and patient's privacy is to be ensured, and the quality and the integrity of the data have to be guaranteed. A widely used mechanism to meet these requirements is Access Control (AC). In a EHR-S the AC is used to limit the access and to indicate how and who is allowed to operate on the clinical documents. This work presents an advanced access control model for the EHR-S. It starts from another AC model proposed by the authors, namely MPP-ABAC model [2], extending several its characteristics, and focusing major attention on assuring privacy for patients. The novelty of our work is the use of the principle of least privilege (POLP) [3], applied to the information content of clinical documents. The principle of least privilege, also known as the principle of minimal privilege, is the practice of limiting access to the minimal level that allows normal functioning. The principle applied to the information content of documents translates to increase the confidentiality and therefore to indicate the only information strictly necessary for the given user. Any other information is hidden from the user. The new defined access control model (AC model) introduces new components and support functionalities, through which the patient is able to create a view on clinical document and to specify the lists of able users and not able users. In this way the patient can choose the parts of a document (the sections) that a given user is able to access. The major fine-grained definition of privacy policies allows patients to increase a certain degree of trust in the system. This feature has an important impact on the social dynamics; as a matter of fact, the patient is more encouraged to share a clinical document. In a EHR-S are numerous clinical documents in EHR systems that contain very sensitive information (such AIDS, etc..), so the patient is encouraged to avoid the sharing of the full document to specific healthcare users. With the definition of the View on documents, the patient can share only the less sensitive parts (sections) to the specific healthcare users.

2 Related Work

In this section, we will briefly survey Access Control (AC) model related works. Over the last few years, different models, which aim at satisfying the needs for the protection and the privacy of sensitive, have been defined. The ones most frequently used for Electronic Health Records [4], are the Role Based Access Control model (RBAC) [5], [6] and the Attribute Based Access Control model (ABAC) [7]. The RBAC model, uses the concepts of users, operations and groups, and defines the concept of role. It grants or denies access to certain operations depending on the role of the user. The ABAC model controls access to objects by evaluating rules against the attributes of the entities (subject and object), actions and the environment relevant to a request. Another model, which focuses on the need for definitions of policy related to the requirements of patient privacy, is the Privacy-Role Based Access Control model (P-RBAC) [8]. This model is an extension of the RBAC model, which not only the role and the permissions that such a role has on the required object are considered, but also the purpose of the access to the object and the defined

privacy policies in compliance with the users' will. Another very flexible model is the Temporal Role-Based Access Control model (T-RBAC) [9], which allows a temporal enabling and disabling of the role. Each of these models responds just partially to the patients' need for privacy, as most of them have limitations in the possibility of accurate and flexible managing the security policies. In our previous paper, we proposed a new access control model to meet the requirements for the patient's privacy in a EHR-S. It is a fine-grained access model in compliance with the main security needs for EHR-S. In modern EHR-S, it is necessary to give directly to the patient the opportunity to manage the policies regarding the access to his/her documents (this is the reason why the MPP-ABAC is defined as "patient privacy-centric"), and also, the European directives [10] move in this direction. Through the components of the model, the patient can easily and dynamically define her/his own security policies, allowing or denying access to her/his documents to specified roles/users and for given purposes. The patient privacy-centric characteristic is introduced into the model through the definition of the: *Purposes, List, Temporal* and *Limitations* components and through the introduction of an additional functionality for a dynamic management of document access. In fact, the patient can choose the purposes he/she wants to associate with each of his/her documents, which are already present in the system. The *List* component enables a dynamic managing of the users and the roles by the patient. It allows a definition of the list of users and/or roles associated with the *Permission* component. In this way, it is possible to specify which users have the permission and which ones do not have permission to access through the definition of the *Able* and *NAble* relationships. The relations between *List, Temporal*, and *Permission* are shown in figure 1, in which the new model is displayed. Our goal in this study is to design an access control model that is patient privacy-centric and provides the users the capability of defining their privacy preferences in a dynamic and fine-grained manner. In this paper, we propose a view-based access control model that can be used for users to fully control their privacy on the clinical documents.

3 Proposed Model

In this section, we describe the proposed access control model.

3.1 Model Overview

The MPP-ABAC model defined in [2] allows a patient to manage the privacy policy of his/her clinical document in a dynamic and fine-grained manner. To the initial model, we have added the View component (green in the figure 1) in order to increase the degree of detail in the definition of privacy policies. The View component, as described in detail below, allows the meeting of the principle of least privilege [3]. In fact, the patient is able to choose parts of his/her clinical document through the View component, and to grant the access of specific users' lists only (through List component and Able and NAble relations). In order to enable the patient to indicate

in a simply and guided manner his/her preferences in terms of privacy for his/her documents in an EHR-S, we have defined several support functionalities. They allow to use the full potential of the model, for example defining View bind to documents and associating the View to the users' lists.

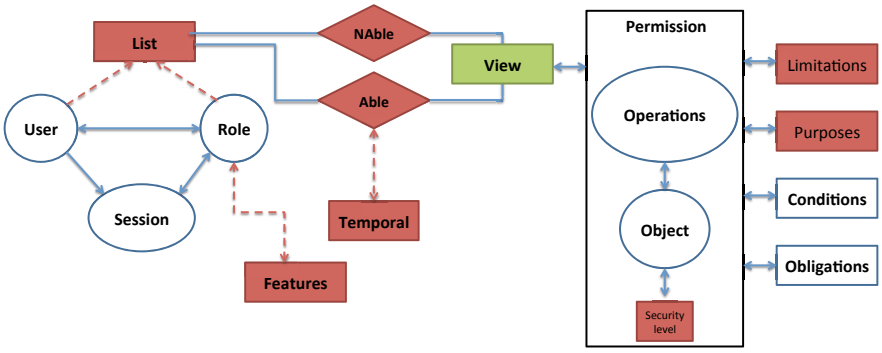


Fig. 1 The figure shows the access control model proposed. In white, there are the RBAC components. In red, there are the MPP-ABAC components, and in green, the component introduced in this paper.

3.2 Model Components

The fundamental components of the MPP-ABAC model are:

- The **Temporal** component, which allows the management of access rights depending on the time condition, as in the Temporal Model-Based Access Control [9].
- The **Permission** component (RBAC model [11]), which allows the specification of the access rights, that is, it defines which operations are permitted on various objects.
- The **List** component, which allows patients to define a list of both role user and id user.
- The **Able** and **NAble** relations. It is possible to indicate the association between system users and permissions on objects by means of the List relation. Therefore, this enables the model to easily indicate the users who have the right to operate on the objects and those who do not have this right.
- The **Purposes** component, which associates the Intended Purposes (that is, the purpose for which a particular document has been collected) to objects, with the goal of limiting document access to the listed purposes only.

The novel component introduced by this work is:

- The **View** component, which is placed in the middle between the Able and NAble relations and the **Permission** component, constituted by object (clinical

document) and by operation (in figure 1). Through this, it is possible to define a "list of parts" that is a view on a clinical document, made of a list of parts (document sections). The patient can associate a users' list with a defined list of parts, through the Able relation. In this way, he/she obtains an *AbleView Association*. Through NAble relation, it is possible to associate a View on document with a users' list. Thus, he/she obtains a *NAbleView Association*.

3.3 Support Functionalities (SF)

In addition to the definition of the access control model, we have defined new support functionalities (SFs). They allow a patient to correctly use the model for the definition of the privacy policies. The support functionalities defined in the initial AC model are described in [2]. In this paper we introduce additional functionalities, which allow the use of the View component. Below, there is a brief description of such new functionalities: *Create View*, *Bind View to List*, *Define View on the Document*.

- **Create View**
  This SF allows the patient to define views on his/her clinical documents (lists of parts). The patient is able to choose the parts of his/her clinical document to create the view. In this way, the patient defines the document view, which represents the clinical document made by only parts indicated by the patient. The support functionality uses the View component of the model.
- **Bind View to List**
  This SF allows the patient to bind a View (that is defined by Create View) to a users list (that is defined by the List component). The patient through this functionality is able to indicate, for a specific users list, the View (or the Views) to which the users (in the list) are able to access. When a user member of the users list requests access to the document, he/she receives only the View to which he/she is authorized to access (the model shows only the portion of the document that the patient has decided to let him/her see). The patient is able to define two different associations, *AbleView* which permits the access and *NAbleView* which denies the access.
- **Define View on the Document**
  This SF allows the creation of the view on the document (another document) starting from the list of parts that the user is able to access. Subsequently, the execution of the control algorithm identifies the list of parts (sections) of the document that the requesting user is able to access to and then creates the view on the document. The View on the Document (that is a document) will be sent to the requesting user.

3.4 Internal Functionalities

The AC model includes also several internal functionalities. One of these is the "create the policy", that allows to translate the patient's will in the privacy policy within the system. This functionality permits the creation of a new policy (that updates the previous policy) avoiding conflicts in control time. The creation of the policy occurs when the patient uses the SF *Bind View to List*. When the patient associates a users' list with a specific view, he/she allows all users in the list to access only the portion of the document in the view. The policy management in our model, during the definition of the security policy itself, expects to minimize the possibility of conflicts (among policies) in control time. This means that when the patient associates the users' list to the View after that he/she created the list and the view, the system performs the following actions:

- **1.** for each user in the list, the AC system checks if the user is in other lists associated (Able relation) with the complete document (View ALL);
- **2.** the system creates a users' list with all the users discovered in step 1;
- **3.** the system creates a view (list of the parts) with all the parts of the document except those in the view created by the patient;
- **4.** the system binds the list created in step 2 with view created in step 3, via NAbleView relation;
- **5.** the system binds the list created by the patient with the view created by the patient through the AbleView relation.

4 Implementation of the Control

A specific algorithm, shown in the activity diagram (figure 2), realizes the access control. It consists of a series of control functions, which are:

- *CheckAble*
  checks if the user that requires a specific document is associated with an Able list (the user can access the complete document).
- *CheckNAble*
  checks if the user that requires a specific document is associated with a NAble list (the user cannot access to any part of the document).
- *CheckAbleView*
  checks if the user that requires a specific document is associated with an AbleView list (the user is able to access some parts of the document).
- *CheckNAbleView*
  checks if the user that requires a specific document is not able access to any parts of the document.

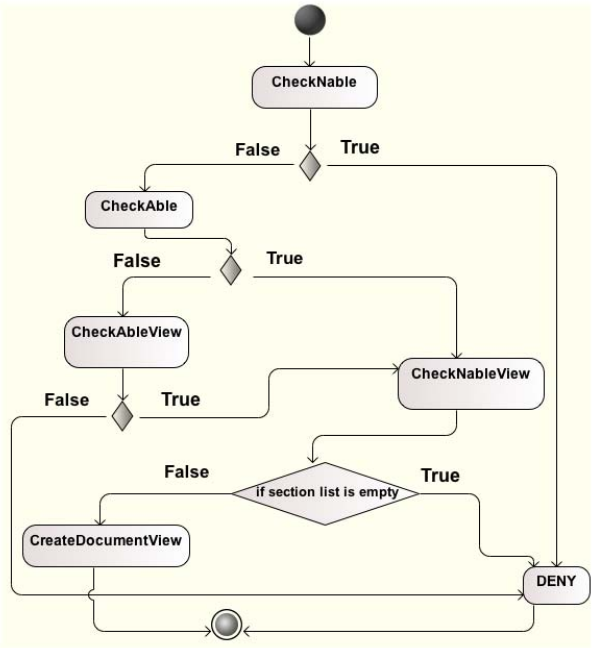


Fig. 2 Activity Diagram of the control algorithm for View-based MPP-ABAC model

5 Running Scenario

The AC model defined in this paper has been evaluated in a specific running scenario, using a very critical clinical document, the *Patient Summary* (PS); it is a document that collects the patient's clinical information of more interest, and it is very useful, for example, in case of emergency. The PS typically is structured according to the HL7 CDA rel. 2 [12], consists of several sections, some obligatory and others optional. For example, the sections "Problem List" and Allergies are mandatory. The carried experimentation has highlighted both the characteristic of flexibility and the simplicity of the management of the policy via View. The experimentation showed how the principle of least privilege affects clinical and social dynamics. In order to analyze how the model works and how it manages privacy policies according to the needs expressed by the patient, this section presents a running scenario of the AC model in which the patient wants to make certain parts of a document accessible uniquely from a set of healthcare users.

Let us consider the following starting situation (shown in figure 3 a)

- The users in the List 1 (namely the General Practitioner (GP) and the Orthopedic), can access all the parts (sections) of the document (in fact, this list is connected with the relation Able to the "View ALL" view);

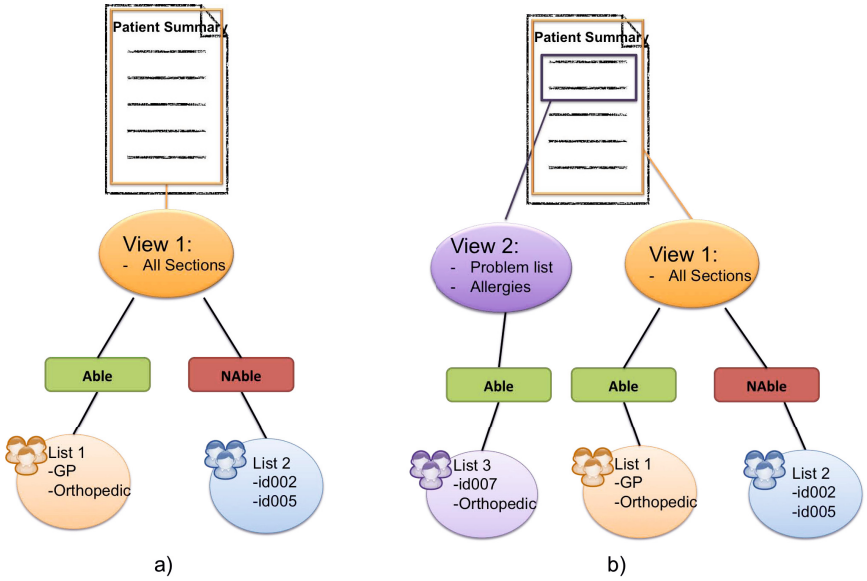


Fig. 3 Graphical representations of the relations among the Patient Summary, the views and the users lists

- The users with id002 and id005 identifiers in the List 2 are not able to access the complete document, in fact the list is bonded with NAble relation to the "View ALL" view).

After that, the running scenario expects that the patient wants to make the sections "Problem List" and "Allergies" of his/her PS accessible both to given user (for example his/her dentist) and to the orthopedic role (for orthopedics, the access is available only for the Problem List and Allergies sections). For this reason, the patient follows the steps below:

- The patient creates a new users list (List 3), which is composed both by the orthopedic role and by the identifier of the patient's dentist (in this case, id007);
- The patient creates a new view on the document, which is composed by the "Problems List" and "Allergies" sections;
- The patient associates the List 3, via the Able relation, with the view created in the previous step (shown in figure 3 b).

As already mentioned in subsection 3.4, when the patient creates the association between list and relation, the AC model uses the internal functionality "create the policy" and defines the privacy policies that meet the patient's will (avoiding conflicts among policies). In this specific scenario, the starting situation is in conflict with the patient's will, because, at the beginning, the orthopedic role is able to access to the complete document, but the patient now indicates that the orthopedic role can only access to the sections "Allergies" and "Problems List". The "create the policy" functionality executes the following steps:

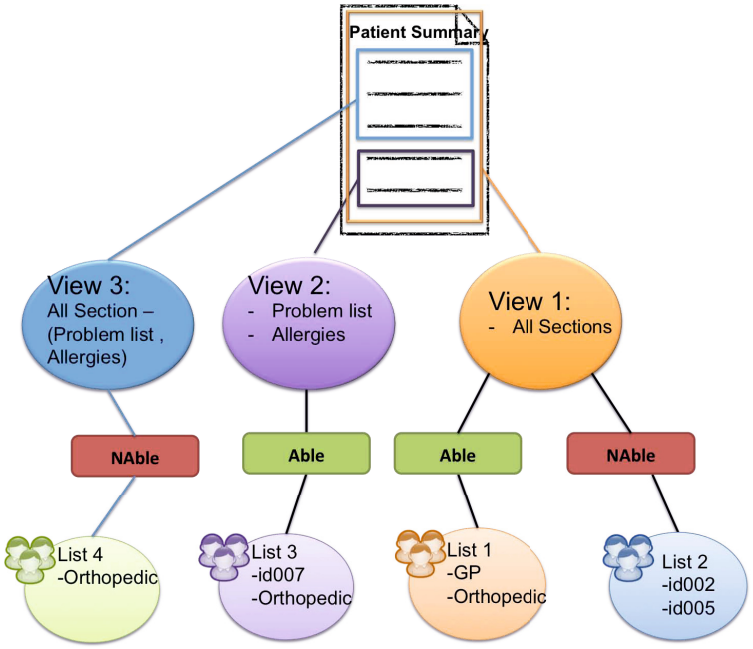


Fig. 4 Final graphical representation of the relations among the Patient Summary, the views and the users lists in the running scenario

- **1.** creates a list with the orthopedic role (Orthopedic was associated with the Able list);
- **2.** creates the view with all the sections except the Problem List and Allergies sections (ALL-(Problems List and Allergies));
- **3.** creates the NAbleView relation between the view created in step 2 and the list created in step 1.

In this way, the privacy policy outcome is not in conflict with other existing policies (in respect to the control algorithm presented in figure 2). This final situation is shown in figure 4.

6 Conclusion and Future Work

In this paper, we have proposed a View-based approach that, inheriting the characteristics from the MPP-ABAC model, meets the principle of least privilege for the clinical documents. In the philosophy of the principle of least privilege, it is necessary to provide only the information strictly necessary for the given healthcare user, and any other information is hidden from the user. In the work, we have presented an access control model, as well as an algorithm for the implementation of the con-

trol, which is able to eliminate the conflict among privacy policies. On one hand, future work is directed to evaluate the proposed model in a real system for EHR (for example in the InFSE EHR infrastructure [13]), on the other hand, to introduce new intelligent systems aiming at providing the minimum necessary information to certain requests by specific healthcare users. For example, such systems could permit a doctor, instead of asking for a list of allergies of a patient, to ask if the patient is an allergy to a specific allergen.

References

1. Kilic, O., Dogac, A.: Achieving Clinical Statement Interoperability Using R-MIM and Archetype-Based Semantic Transformations. IEEE Transactions on Information Technology in Biomedicine 13(4), 467–477 (2009), doi:10.1109/TITB.2008.904647
2. Sicuranza, M., Esposito, A.: An Access Control Model for easy management of patient privacy in EHR systems. In: In the 8th International Conference for Internet Technology and Secured Transactions, ICITST-2013 (2013)
3. Schneider, F.B.: Least Privilege and More, http://www.cs.cornell.edu/fbs/publications/leastPrivNeedham.pdf (access date: February 14, 2014)
4. Sicuranza, M., Ciampi, M., De Pietro, G., Esposito, C.: Secure Healthcare Data Sharing among Federated Health Information Systems. International Journal of Critical Computer-Based Systems 4(4), 349–373 (2014), doi:10.1504/IJCCBS.2013.059023
5. Sandhu, R., Ferraiolo, D., Kuhn, R.: The NIST Model for Role-Based Access Control: Towards A Unified Standard (2000), http://csrc.nist.gov/rbac/sandhu-ferraiolo-kuhn-00.pdf (access date: July 11, 2013)
6. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-Based Access Control Models. Computer 29, 38–47 (1996), http://dx.doi.org/10.1109/2.485845, doi:10.1109/2.485845
7. Shen, H.-B., Hong, F.: An Attribute-Based Access Control Model for Web Services. In: Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT 2006, pp. 74–79 (December 2006), doi:10.1109/PDCAT.2006.28
8. Kim, Y., Song, E.: Privacy-Aware Role Based Access Control Model: Revisited for Multi-Policy Conflict Detection. In: 2010 International Conference on Information Science and Applications (ICISA), April 21-23, pp. 1–7 (2010), doi:10.1109/ICISA.2010.5480349
9. Bertino, E., Bonatti, P., Ferrari, E.: TRBAC: a temporal role-based access control model. In: Proceedings of the ACM Workshop on Role-Based Access Control, pp. 21–30. ACM Press, New York (2000)
10. General Data Protection Regulation, European Commission, regulation of the european parliament and of the council (2012), http://ec.europa.eu/justice/data-protection/document/review2012/com_2012_11_en.pdf (access date: September 13, 2013)
11. Ferraiolo, D.F., Cugini, J., Kuhn, D.R.: Role-Based Access Control (RBAC): Features and Motivations. In: Proceedings of the 11th Annual Computer Security Application Conference, New Orleans, LA, December 11-15, pp. 241–248 (1995)
12. HL7 Version 3 Clinical Document Architecture (CDA) Release 2, https://www.hl7.org/implement/standards/product_brief.cfm?product_id=7 (access date: March 24, 2014)
13. Ciampi, M., De Pietro, G., Esposito, C., Sicuranza, M., Donzelli, P.: On Federating Health Information Systems. In: Proceedings of the International Conference Healthcare Informatics and Biomedical Engineering, HiBES (July 2012)

Resilient Semantic Sensor Middleware

Gianpio Benincasa, Giuseppe D'Aniello, Matteo Gaeta,
Vincenzo Loia, and Francesco Orciuoli

Abstract. Resilience is the capability of a system to absorb and mitigate unexpected faults and risks. This paper describes the definition of a resilient middleware for sensor network management in dynamic environments for supporting Situation Awareness processes in security scenarios.

The proposed approach is based on Quality of Service of sensors for identifying faults, disturbances, incompleteness and uncertainty in raw data. Moreover, Dempster-Shafer Theory is employed to aggregate sensor data and abstract them to obtain coherent observations about the monitored environment. Lastly, machine learning techniques are used to discover association rules in order to handle the absence of relevant observations.

1 Introduction and Related Works

Situation Awareness has been defined by Endsley as *"the perception of the elements in an environment within a volume of time and space, the comprehension of their meaning, and a projection of their status in the near future"* [3]. Situation Awareness represents a critical aspect for improving decision-making processes in several and heterogeneous domains like, for instance, Security (both cyber and physical), Emergency Management, Energy Savings, Logistics, and so on. One of the most important steps in Situation Awareness is Situation Identification that is a challenging task. In literature, numerous approaches for Situation Identification [12] are recognizable. In the Endsley model, perceiving context information means abstracting

Gianpio Benincasa · Giuseppe D'Aniello · Matteo Gaeta · Francesco Orciuoli
DIEM - University of Salerno, Via Giovanni Paolo II, 132. Fisciano (SA) Italy
e-mail: {gbenincasa,gidaniello,mgaeta,forciuoli}@unisa.it

Vincenzo Loia
DI - University of Salerno, Via Giovanni Paolo II, 132. Fisciano (SA) Italy
e-mail: loia@unisa.it

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_47

data coming from a set of sensors and sensor networks deployed in order to observe a specific environment. However, sensors are devices prone to failure or malicious attacks, which can generate anomalies in data and impact on the fairness of Situation Identification task. In order to face these cases, it needs a suitable level of resilience. A resilient control system is one that maintains state awareness and an adequate level of operational normality in response to disturbances, including threats of an unexpected and malicious nature [10]. For our interpretation, *resilience* is the capability to absorb and mitigate faults and challenges to normal operation by continuing to provide and maintain an acceptable quality of service. Faults and challenges are intended both as tampering or breakage of sensors, both as uncertainty or unreliability of information. This paper provides the definition of a resilient middleware for sensor network management to support Situation Awareness processes in decision-making scenarios. The work starts from existing solutions, considers their gap with respect to some characteristics related to resilience and exploits Semantic Technologies and Computational Intelligence techniques in order to fill the above mentioned gap. The proposed middleware, leveraging on a multi-agent architecture, exploits the concept of Service Level Agreement (SLA) to detect anomalies in sensor data and uses techniques for learning association rules (e.g., the approaches proposed in [2] and [5]), to handle cases in which some Observations, provided by sensors, are absent or unreliable. The result is that the provided middleware, in cases of faults of some sensors, is able to continue supporting the Situation Identification task and, consequently, the decision-making processes. In literature, several works providing different solutions for managing sensor networks have been recognized. Among the most important ones, the authors of [1] propose the Global Sensor Network (GSN), a sensor network middleware. GSN allows users to reconfigure the running system, to add new sensor *on-the-fly* and to monitor the effect of the changes via a graphical interface but it does not provide automatic mechanisms for adding and removing sensors as well as the approach proposed by Ganz et. al [4] who uses semantic data annotations. Moreover, CA4IOT [8] is an architecture designed to automatically select sensors according to the current users' needs. CA4IOT allows the users to build high-level aggregation of the sensor data but it does not provide any mechanisms for managing the dynamism of the environment. The authors of CA4IOT also propose CASCoM [9], a model addressing the challenge of providing automatic context-aware configuration mechanisms for filtering, fusion, and reasoning in IoT middleware. CASCoM is also based on GSN and proposes a semantic layer based on SSN. Lastly, Le-Phuoc et al. [6] introduce Linked Stream Middleware (LSM), a platform that brings together the real world sensed data and the *Semantic Web*. LSM provides several functionalities for data manipulation (e.g., data collection, annotation, querying, etc.). However, also LSM does not provide mechanisms to deal with dynamic ecosystems. Definitely, three main relevant lacks emerge when the above existing solutions are analysed and compared with respect to the challenge of providing a comprehensive resilient sensor network middleware. The first one, i.e., *Dynamic Environment Management*, concerns the capability to react to the changes occurring in the environment (e.g. a sensor failure, etc.); the second one, i.e., *Semantic Data and Aggregation*, is related to the ability of modelling the sensor data and the surround-

ing environment in order to integrate several sensors and provide information about Quality of Service (QoS); the third one, i.e., *Situation Awareness Support*, is useful for abstracting raw data coming from the measurements and producing high level Observations.

2 The Proposed Semantic Middleware

The role of the proposed Semantic Middleware is to support people and applications, which are involved in decision making processes, with high-level *perceptions* related to the Situations occurring in the monitored environment. Fig. 1 shows the different levels of abstraction which are considered in this work. In particular, starting from the lowest layer, it is possible to find: i) *Data* provided by sensors and sensor networks; ii) *Observations* obtained as result of a first processing step; iii) *Context Attributes* which are more abstract information provided by some processing tasks applied on Observations; iv) *Situations* which represent the highest level of generalization of context information and are obtained by processing Context Attributes. This work mainly focuses on the two middle levels of the pyramid in Fig. 1 by providing effective evaluation of Context Attributes also in case of errors and anomalies recognised in sensor data. The proposed approach is *perception-based* in the sense that it borrows from human perception in order to implement the required data abstraction operations.

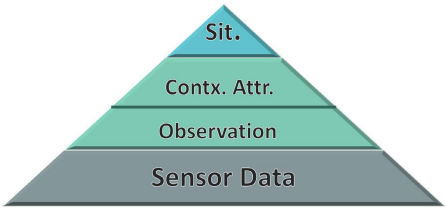


Fig. 1 Informative Architecture

The *Semantic Middleware*, as depicted in Fig. 2, consists of two main components: *Multi-Agent System* and *Semantic Model*. The Semantic Model describes information related to sensors, measurements and environment by means of the Semantic Web stack[1] in order to support interoperability, cooperation and aggregation of data. Furthermore, the Semantic Model considers concepts like QoS and SLA related to each sensor for supporting quality assessment of sensor measurements. The main capability of the Multi-Agent System is to observe the environment (by means of sensors) and to represent its dominant features by means of Context Attributes. Context Attributes are used for identifying the current Situation that will be used to

[1] W3C Semantic Web Specifications `http://www.w3.org/2001/sw/Specs`

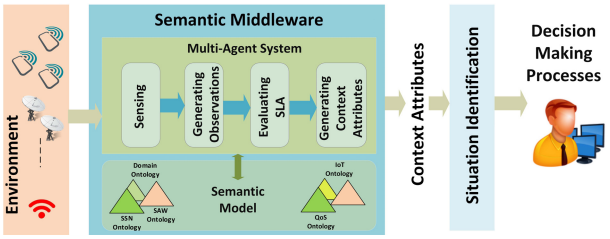


Fig. 2 Semantic Middleware

support human operators in making decisions in agreement with the current environment state. The Context Attribute identification process consists of four phases.

The *Sensing* phase is responsible for gathering sensor measurements. The middleware contains one agent for each sensor of the environment. Let us consider that more than one sensors are deployed in the environment for observing the same characteristic. For instance, more than one security camera can be deployed in the environment for observing the presence of people. Raw data gathered by a sensor is transformed in a semantic representation by an agent, with respect to the ontologies of the Semantic Model. The *Generating Observations* phase is responsible for identifying the Observations related to specific environmental features. One agent is deployed for each feature of the environment that has to be evaluated. For instance, an Observation Agent is responsible for evaluating the presence of people (e.g. by considering data coming from cameras). An agent collects data from several sensors (of the same type) which are deployed in order to observe the same phenomena. The goal of the agent is to identify the best measurement among all observed sensors (with respect to the current QoS of each sensor), by combining their measurements. The output of this phase consists of the Observation value and an evaluation of its quality. In the *Evaluating SLA* phase, the quality of each Observation is compared with a minimum level of acceptance for that feature, with respect to the correspondent SLA threshold. In our proposal, SLA is a threshold that represents the minimum value of quality for accepting an Observation. A SLA threshold has to be defined for each kind of Observation. The output of this phase indicates if an Observation can be used by the Semantic Middleware for producing useful Context Attributes. The *Generating Context Attributes* phase produces Context Attributes by exploiting the Observation values. Moreover, this phase is responsible for identifying the Observations which can be used in the place of the ones that does not overcome their SLA thresholds. Association rules are used for identifying those Observations.

Definitely, the proposed architecture exploits semantic technologies for representing relevant knowledge allowing interoperability and cooperation among software agents. Moreover, the middleware exploits techniques based on Dempster-Shafer Theory (DST) for aggregating data of different sensors (Data Aggregation) in order to support their representation with higher-level, domain-specific concepts

(Observations and Context Attributes). Sensor data are processed for being represented in domain concepts with a growing level of abstraction (see Fig. 1); the highest level (Context Attributes) supports Situation Identification, which is the most challenging task of Situation Awareness. Lastly, the middleware exploits an approach based on QoS and SLA for evaluating the quality of service of each sensor for dealing with dynamic environments and uncertain and unreliable information. The middleware employs machine learning techniques for generating association rules. These rules allow replacing missing information with other Observations.

2.1 Semantic Model

The Semantic Model is defined as a set of integrated OWL[2] ontologies, deployed into an RDF Storage System, and it consists of two main parts. The first part, *Sensors, Observations and Environment (Fig. 3-a)*, concerns the description of sensing devices, their characteristics, their measurements and other environmental features. Sensing devices and their measurements are described by means of the *Semantic Sensor Network Ontology*[3] (SSN). The environmental information, instead, describes the surrounding environment: position of the sensor, time, weather, pressure, physical phenomena and so on. Environmental information are represented by means of several domain-specific ontologies.

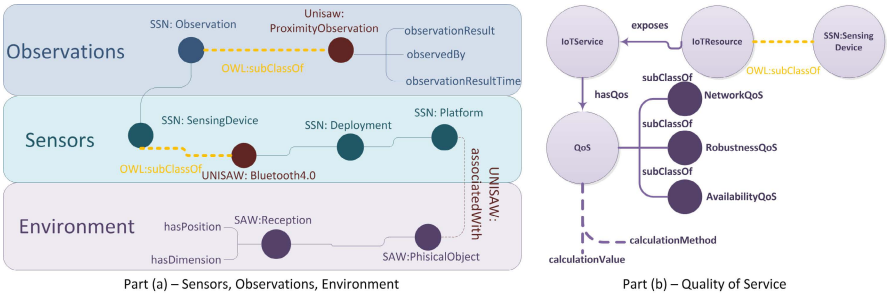


Fig. 3 Semantic Model

The second part (*Quality of Service*) (Fig. 3-b) allows to describe the parameters which contribute to the definition of the quality of a sensor measurement (e.g., packet loss rate, throughput, jitter, delay, and so on). These concepts are modelled by using the ontologies proposed in [11]. Each kind of sensor has its own QoS parameters which are represented as sub-classes of `QualityOfService`. The value

[2] OWL `http://www.w3.org/TR/owl2-overview/`

[3] Semantic Sensor Network Ontology `http://www.w3.org/2005/Incubator/ssn/ssnx/ssn`

of each QoS parameter is specified by means of the `calculationValue` property. Moreover, the property `calculationMethod` can be used for representing an expression that describes a method for calculating the value of a specific QoS parameter.

2.2 Generating Observations and Evaluating SLA

Fig. 4 depicts the process for generating Observations and evaluating SLA.

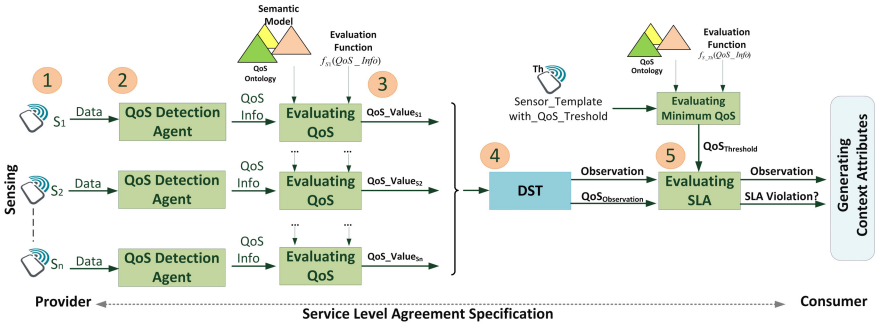


Fig. 4 Generating Observations and Evaluating SLA Process

Let us consider an environment in which n sensors of the same kind are deployed in order to observe the same environmental feature (e.g., n cameras for observing if someone is in a room). These sensors (step 1 in Fig. 4) produce measurement data that will be used for determining a value for a specific Observation. An agent, namely *QoS Detection Agent*, gathers the information needed for evaluating QoS parameters described in the Semantic Model (step 2). For instance, the agent acts as a *sniffer* on the network in order to evaluate packet loss ratio, throughput, jitter, delay, and so on. These parameters are used in the next step for evaluating the overall QoS of each measurement (step 3). The overall value of QoS is needed for comparing and combining the measurements of the sensors in order to determine the Observation value. The overall value of QoS of a $Sensor_i$ is calculated by means of an evaluating function that is specific for each kind of Observation:

$$Overall\_QoS_{Observation_k}(Sensor_i) = \sum_j \alpha_j \cdot f_j(QoS\_Parameter_j) \quad (1)$$

where $QoS\_Parameter_j$, with $j = 1..m$, are the m QoS parameters of $Sensor_i$ related to the k^{th} kind of Observation ($Observation_k$). The contribution of j^{th} parameter to the overall value of QoS depends on the weight α_j and on the function f_j (that is specific for each parameter and allows to transform a parameter into a value that is comparable to the other values).

The overall values of the QoS and the values of measurements of each sensor are combined for determining the Observation value by exploiting the DST theory (step 4). DST is a mathematical theory of evidence based on belief functions and plausible reasoning, which is used to combine separate pieces of information (evidence) to calculate a degree of belief (the probability) of an event, by taking all available evidences into account. In a DST reasoning scheme, the set of admissible hypotheses Θ, namely the *frame of discernment*, represents the set of choices $\{h_1, h_2, ...h_n\}$ available to the reasoning scheme [7]. The sources (e.g. the sensors) assign a belief or evidence across the frame hypotheses. Let 2^{Θ} denote the set of all subsets of Θ to which a source of evidence can apply its belief. Then the function $m : 2^{\Theta} \rightarrow [0, 1]$ is the *mass function* that defines how *belief* is distributed across the frame. The main important feature of the process for assessing evidence in DST is the ability to combine evidence from multiple sources. The combination of evidence from two independent sources (called the *joint mass*) is accomplished by *Dempster's combination rule*:

$$m_{1,2}(A) = \frac{1}{1 - \sum_{X \cap Y = \varnothing} m_1(X) m_2(Y)} \sum_{X \cap Y = A} m_1(X) m_2(Y) \qquad (2)$$

$m_{1,2}(A)$ is the combined belief for a given hypothesis A. Let us explain how DST is exploited in order to calculate Observation values. For every kind of Observation (e.g. presence of people in a room, temperature of the room, etc.), a set of hypotheses $\Theta = h_1, h_2, ...h_n$ is defined (e.g., h_1=absence of people in the room, h_2=presence of people in the room). The goal of DST is to determine which hypothesis is true and which is the value of belief. To achieve this, the agent defines the belief of each sensor across all the hypotheses by means of a *mass function* that depends on the QoS parameters of the sensor. Belief from the source of evidence (i.e. sensor data) is fused in order to determine total belief of the Observation, using *Dempster's combination rule*. The hypothesis with the highest belief represents the value of the Observation. Section 3 provides an example of generating Observations with DST.

Once an Observation value has been calculated, its quality has to be evaluated with respect to the SLA (step 5). The quality of the Observation value is given by a linear combination of all the QoS values of sensors, by considering only those sensors which have stated exactly the value that has been selected by DST. Then, this value is compared with a minimum threshold that indicates the lowest level of quality that does not violate the SLA. A SLA specification contains the minimum values of each QoS parameter that are required by a service consumer. For each kind of Observation, the SLA is specified by means of a *Sensor Template*. The Sensor Template is a semantic specification of a sensor that is compliant to the SLA of an Observation. The quality of an Observation is compared to the minimum acceptable quality of the related sensor template, according to the following definition:

$$SLA(x) = \begin{cases} compliant, Overall\_QoS_{Obs_k}(x) \geq Overall\_QoS_{Obs_k}(Sensor\_Template_k) \\ violated, Overall\_QoS_{Obs_k}(x) < Overall\_QoS_{Obs_k}(Sensor\_Template_k) \end{cases} \qquad (3)$$

$SLA(x)$ indicates the evaluation of the SLA for sensor x and x identifies the sensor that has produced the value for Observation Obs_k (when the value of obs_k is calculated by combining several sensors, x corresponds to a dummy sensor obtained by calculating the mean values of their QoS parameters). The Observation is compliant to the SLA if the quality of its sensor is greater than the quality of the correspondent Sensor Template.

2.3 Generating Context Attributes

The set of Observations respecting the SLA values are transformed into Context Attributes by means of suitable *classification rules*. These rules can be learned by means of supervised machine learning techniques. As stated in Section 1, one of the main weaknesses of existing sensor middleware is the lack of support for dynamic environments in which faults and threats may arise. A resilient middleware, instead, should maintain an adequate level of operational normality in response to these faults which could be malicious. So, when some Observations are missing (due to the aforementioned issues), the proposed middleware can use machine learning approaches to derive other kind of Observations allowing it to identify the same Context Attributes. For instance, if an Observation related to a camera does not respect the associated SLA, the middleware may use a sound-level sensor to detect if someone is in the room. In order to detect which Observation can be used in the place of the missing one, the middleware exploits a set of *association rules*. Fig. 5 shows some examples of association rules. These rules are generated by applying machine learning algorithms to a training set consisting of associations among Observations and Context Attributes. So, when an Observation is missing or its QoS is too poor with respect to the associated SLA, the proposed middleware identifies other Observations by exploiting the above rules. Specifically, when an Observation that is specified in the $conclusion$ of a rule is missing, the middleware can use the Observation(s) of the $premise$ for deducing the same information. Several association rule mining techniques (e.g. A-Priori or FPGrowth) can be used to learn association rules by processing a training set [13]. When dealing with uncertain information or noisy data, it is possible to use *fuzzy association rules*, which can be learned with the approach proposed in [5] or by exploiting Fuzzy Formal Concept Analysis [2].

#1. Obs1 Obs2 Obs3 [*Confidence*=100%]=> <*Support*=1> Obs4;
#2. Obs1 Obs2 Obs4 [*Confidence*=100%]=> <*Support*=1> Obs3;
#3. <*Support*=3> Obs1 Obs4 [*Confidence*=67%]=> <*Support*=2> Obs3;
#4. Obs1 Obs3 [*Confidence*=67%]=> <*Support*=2> Obs4;
...

Fig. 5 Association rules

3 Sample Scenario: Intrusion Detection

In order to evaluate the applicability of the proposed approach, let us consider the following *Intrusion Detection* scenario (see Fig. 6). Intrusion detection is related to security mechanisms for preventing unauthorized access to resources and/or data. Let us consider a room with restricted access, monitored by three cameras and a microphone. Moreover, let us consider that cameras have been tampered with. This implies a degradation in the quality of the received information. In this scenario, two values for an Observation, namely `intrusion_detected`, are available: i) `A=presence_of_intruder` and ii) `B=absence_of_intruder`. The used sensors produce measurements modelled in accordance to the Semantic Model (step 1). The quality of each measurement is determined by means of evaluating functions (step 2) which consider packet loss rate and delay in transmitting information. The overall value of QoS of each sensor is reported in the table (step 3). Furthermore, this table reports the measurement value for each camera and the value for the related Observation. Specifically, camera C_1 asserts `A` while C_2 and C_3 assert `B`. In step 4, the value of each sensor is combined by means of the Dempster's combination rule, in order to obtain the joint mass for hypothesis `A` and `B`. Let us underline that, although QoS of C_1 is higher than C_2 and C_3, the value for Observation is `B` (this is due to the combination of evidences from multiple sources). In the next step (5), the quality of the Observation has to be compared with the SLA. Let us suppose that the QoS of the Observation is lower than the SLA. In this case, the middleware exploits the learned association rules for identifying possible Observations which can substitute the previous one. By employing rule #1 (step 6), the middleware infers that it can use the installed microphone for detecting if someone is in the room. Usually, the use of the microphone for identifying people is less accurate than cameras. But in this scenario, due to the tampering of the cameras, the middleware can use the microphone to compensate for the lack of information. In the step 7, a new Context Attribute, indicating that an intruder is present in the room, is generated by considering an Observation derived by the measurements of the installed microphone.

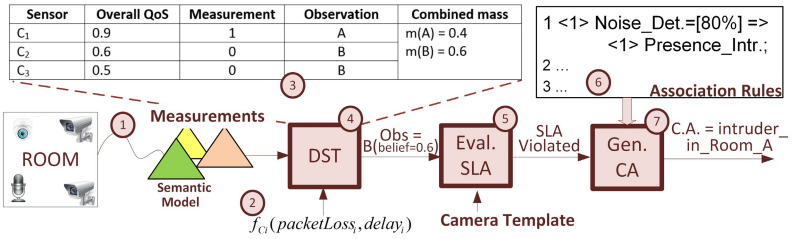


Fig. 6 Sample Scenario

4 Final Remarks

This paper proposes a resilient sensor middleware for supporting Situation Awareness for decision making processes in dynamic environments. The description of the middleware is organized into two parts: i) a Semantic Model that describes information related to sensors, environment and QoS, by using semantic technologies in order to support interoperability and cooperation among agents, and ii) a Multi-Agent System that observes the environment (by employing sensors) and represents its dominant features by means of Context Attributes. The Multi-Agent System exploits the Dempster-Shafer Theory for abstracting and aggregating sensor data and machine learning techniques for mining association rules. These aspects enable the compensation for faults, disturbances, threats and uncertain information related to the environment. As shown in Section 3, the middleware provides promising results in security scenarios.

References

1. Aberer, K., Hauswirth, M., Salehi, A.: A middleware for fast and flexible sensor network deployment. In: Proceedings of the 32nd International Conference on Very Large Data Bases, VLDB 2006, pp. 1199–1202. VLDB Endowment (2006)
2. De Maio, C., Fenza, G., Loia, V., Senatore, S.: Hierarchical web resources retrieval by exploiting fuzzy formal concept analysis. Information Processing & Management 48(3), 399–418 (2012), doi: http://dx.doi.org/10.1016/j.ipm.2011.04.003
3. Endsley, M.R.: Toward a theory of situation awareness in dynamic systems. Human Factors: The Journal of the Human Factors and Ergonomics Society 37, 32–64 (1995), doi:10.1518/001872095779049543
4. Ganz, F., Barnaghi, P., Carrez, F., Moessner, K.: Context-aware management for sensor networks. In: Proceedings of the 5th International Conference on Communication System Software and Middleware, COMSWARE 2001, pp. 6:1–6:6. ACM, New York (2011)
5. Hong, T.P., Lee, Y.C.: An overview of mining fuzzy association rules. In: Fuzzy Sets and Their Extensions: Representation, Aggregation and Models, pp. 397–410. Springer, Heidelberg (2008)
6. Le-Phuoc, D., Quoc, H., Parreira, J., Hauswirth, M.: The linked sensor middleware–connecting the real world and the semantic web. Tech. rep., Semantic Web Challenge 2011 (2011)
7. McKeever, S., Ye, J., Coyle, L., Dobson, S.: Using dempster-shafer theory of evidence for situation inference. In: Barnaghi, P., Moessner, K., Presser, M., Meissner, S. (eds.) EuroSSC 2009. LNCS, vol. 5741, pp. 149–162. Springer, Heidelberg (2009)
8. Perera, C., Zaslavsky, A., Christen, P., Georgakopoulos, D.: Ca4iot: Context awareness for internet of things. In: Proceedings of the 2012 IEEE International Conference on Green Computing and Communications, GREENCOM 2012, pp. 775–782. IEEE Computer Society, Washington, DC (2012)
9. Perera, C., Zaslavsky, A.B., Compton, M., Christen, P., Georgakopoulos, D.: Semantic-driven configuration of internet of things middleware. CoRR abs/1309.1515 (2013)
10. Rieger, C.G., Gertman, D.I., McQueen, M.A.: Resilient control systems: next generation design research. In: 2nd Conference on Human System Interactions, HSI 2009, pp. 632–636. IEEE (2009)

11. Wang, W., De, S., Toenjes, R., Reetz, E., Moessner, K.: A comprehensive ontology for knowledge representation in the internet of things. In: 2012 IEEE 11th International Conference on rust, Security and Privacy in Computing and Communications (TrustCom), pp. 1793–1798. IEEE (2012)
12. Ye, J., Dobson, S., McKeever, S.: Situation identification techniques in pervasive computing: A review. Pervasive Mob. Comput. 8(1), 36–66 (2012)
13. Zhang, M., He, C.: Survey on association rules mining algorithms. In: Luo, Q. (ed.) Advancing Computing, Communication, Control and Management. LNEE, vol. 56, pp. 111–118. Springer, Heidelberg (2010)

Use of the Dempster-Shafer Theory for Fraud Detection: The Mobile Money Transfer Case Study

Luigi Coppolino, Salvatore D'Antonio, Valerio Formicola,
Carmine Massei, and Luigi Romano

Abstract. Security Information and Event Management (SIEM) systems are largely used to process logs generated by both hardware and software devices to assess the security level of service infrastructures. This log-based security analysis consists in correlating massive amounts of information in order to detect attacks and intrusions. In order to make this analysis more accurate and effective we propose an approach based on the Dempster-Shafer theory, that allows for combining evidence from multiple and heterogeneous data sources and get to a degree of belief that takes into account all the available evidence. The proposed approach has been validated with the respect to a challenging demonstration case, namely the detection of frauds performed against a Mobile Money Transfer service. An extensive simulation campaign has been executed to assess the performance of the proposed approach and the experimental results are presented in this paper.

1 Introduction

Frauds in the field of electronic payments continuously evolve as new payment technologies and platforms are introduced. Mobile Money Transfer (MMT) refers to payment services which allow to use virtual money in order to carry out payments, money transfers, and transactions through mobile devices. Such services are being increasingly adopted all over the world, particularly in developing countries where banking services and infrastructures are not so largely available as in developed countries and MMT solutions are being deployed to provide payment services to the so-called "'unbanked"' or "'underbanked"' people. Like any other money transfer service, this service is exposed to the risk of money laundering, i.e., the misuse

Luigi Coppolino · Salvatore D'Antonio · Valerio Formicola · Carmine Massei · Luigi Romano
University of Naples Parthenope, Department of Engineering Naples, Italy
e-mail: {luigi.coppolino,salvatore.dantonio,valerio.formicola,
lrom}@uniparthenope.it, carmine.massei@gmail.com

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_48

consisting in disguising the proceeds of crime and illegal activities and transforming them into ostensibly legitimate money or other assets, or more generally to fraud risks that imply any intentional deception performed to gain financial profit. In this paper we propose a fraud detection system that relies on the Dempster-Shafer theory to spot evidence of ongoing security attacks against MMT systems. This theory is a data fusion technique that allows to combine multiple evidences and to compute a belief value.

The paper is organized as follows. Section II gives an overview of data fusion techniques, with focus on Dempster-Shafer theory; Section III describes the Mobile Money Transfer case study and the frauds considered in this paper; Section IV describes the proposed detection system applied to the MMT case study; in Section V experimental tests and results are shown; Section VI concludes by remarking achieved results and defining future works.

2 Data Fusion Techniques

Data fusion is a process whereby data from multiple sources are combined to yield improved accuracy and more inferences than those that could be achieved using a single source of information. Historically, data fusion has been used in military applications, like remote sensing [16] and target tracking [14]. Also several civil applications are progressively using data fusion techniques to improve the system security and reliability, like robotics [1], medicine [9] and financial infrastructures [10]. The most important problem in data fusion is the development of appropriate models of uncertainty associated with both the state and the observation process. There exist several methods for representing and reasoning about uncertainty, such as the Dempster-Shafer's Theory of Evidence and the Bayesan Inference. In this paper we used the Dempster-Shafer's Theory of Evidence since we do not have a good knowledge of the probabilistic distribution of the states and therefore we cannot calculate the probability a priori required by the Bayesian Inference. Dempster-Shafer's Theory of Evidence is a mathematical theory of evidence introduced in the 1960's by Arthur Dempster [3] and developed in the 1970's by Glenn Shafer [15]. In the Dempster-Shafer framework a proposition can be seen as subsets of a given set of hypotheses. For example, in a fraud detection system, we can consider the set of hypotheses as the set of categories of frauds. Each anomalous event is a subset of the frame of discernment Θ, hence the propositions of interest are in a one-to-one correspondence with the subsets of Θ. Furthermore the set of all propositions corresponds to the set of all subsets of Θ, which is denoted 2^{θ} and is called power-set. In other words we have a set of possible states of the system $\theta_{1...}\theta_N \in \Theta$ which are mutually exclusive and exhaustive. Our goal is to infer the true system state without having an explicit model of the system, but only relying on some observations $E_1 \dots E_M$. Based upon one evidence E_j we can assign a probability that supports a certain hypothesis H_j; in other words we assign a probability to an element of the power-set. A *basic probability assignment (bpa)* is a mass function m which assigns beliefs to a

hypothesis or, in other words, the measure of belief that is committed exactly to the hypothesis H. Therefore, a basic probability assignment is a function $m : 2^{\theta} \rightarrow [0, 1]$ such that $m(\varnothing) = 0$ and $m(H) \geq 0, \forall H \subseteq \Theta$ and $\sum_{H \subseteq \Theta} m(H) = 1$.

We assign two measures [5]:

- the *Belief* function Bel, describing the belief in a hypothesis H, as: $Bel(H) = \sum_{B \subseteq H} m(B)$. The belief corresponds to the lower bound on the probability or rather measures the minimum uncertainty value about a proposition. Its properties are: $Bel(\varnothing) = 0$ and $Bel(\Theta) = 1$.
- the *Plausibility* function of H, Pl(H), which corresponds to the upper bound on the probability and reflects the maximum uncertainty value about proposition H. The plausibility of H is defined as: $Pl(H) = \sum_{B \cap H \neq 0} m(B)$.

Therefore the true belief in the hypothesis H lies in the interval $[Bel(H), Pl(H)]$, while the degree of ignorance is represented by the difference $Bel(H) - Pl(H)$. The second important part of the Dempster-Shafer theory is a rule of combination that permits to combine two independent evidences $E_1$and E_2 into a single more informative hint:

$$m_{12}(H) = \frac{\sum_{B \cap C = H} m_1(B) m_2(C)}{\sum_{B \cap C = \emptyset} m_1(B) m_2(C)}$$

Based on this formula we can combine our observations to infer the system state based on the values of belief and plausibility functions. In the same way we can incorporate a new evidence and update our beliefs as we acquire new knowledge through observations. The theory of evidence allows to reason with uncertainty based on incomplete and also contradictory information extracted from a stochastic environment. Therefore, such theory does not need to know an "*a priori*" probability distribution on the system states like in the Bayesian approach [8].

3 The Mobile Money Transfer Case Study

The Mobile Money Transfer service is a system where virtual money is used to carry out various types of money transfers and financial transactions. For example, a customer can use his mobile phone to carry out financial operations, such as purchasing goods, receiving salary, paying bills, taking loans, paying taxes or receiving social benefits. MMT systems are experiencing rapid adoption. It is expected that mobile payment systems reach US$ 245B in value worldwide by 2014. At the same time, mobile money users are expected to be 340M, equivalent to 5% of global mobile subscribers [13]. The architecture of a MMT system is shown in Fig. 1.

Three classes of users (i.e., Customer, Retailer of *mMoney*, and Merchant) exist in a MMT scenario. They use their mobile phones to communicate with the operations server. Each user is an *mWallet holder*. An *mWallet* is an account hosted in the system allowing the *mWallet holder* to carry out various operations and transactions by using the *mMoney*. The users are connected to the Operations Server that is in

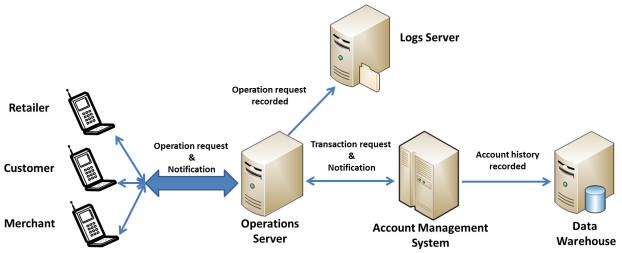


Fig. 1 Architecture of a Mobile Money Transfer system

charge of authenticating the users. It performs simple account management operations (like change the PIN code) and delivers notification messages. It is linked to the Account Management Server which manages the accounts (particularly, if the operation concerns the control of credit/debit). The Account Management Server also stores all the information regarding the user's behaviour.

The Operations Server is also linked to the Logs Server that collects the logs of the various operations that are carried out. The logs contain a wide range of information, such as requests for PIN modification, failed authentication, transaction request, transaction success notification. Historical account management data are stored in the Data warehouse and can be used to analyse customer behaviour. Both log information and account management data can be used to detect frauds. Therefore the input of the system is an operation request received from *mWallet holders*, while the output of the system is the notification of the operation's success/failure, the registration of transaction information and operation information, and the implementation of the requested operation.

Like any other money transfer service, this service is affected by security issues, such as money laundering, privacy protection, frauds, and credit and liquidity risks. Since the success of any payment system is based on ubiquity, convenience, and trust, it is necessary to address emerging risks in order to maintain public confidence in mobile money. To address the security issues of a MMT system, we used the model of such a system developed by the EU FP7 MASSIF ("*MAnagement of Security information and events in Service InFrastructures*"). The MASSIF project has investigated and developed several misuse cases [11]. In order to test the proposed approach we selected the use case named Account Takeover. In this misuse case a fraudster steals the mobile phone from its legitimate user and uses it to perform money transfer. In this misuse case it is very likely that the thief's behaviour differs from the original user's one. Therefore, in order to detect such a misuse case a learning stage is needed. In other terms, the fraud detection system has to be trained by feeding it with information on the user's habits and his usual behaviour. Since

user's data cannot be disclosed due to privacy reasons, we used the MMT simulator developed in the framework of the MASSIF project to generate synthetic data for the learning phase.

4 Fraud Detection through Data Fusion in a MMT System

Fraud detection is the identification of an actual or potential fraud within a system. It relies upon the implementation of appropriate processes to spot the early warning signs of a fraud and can help to uncover new frauds in action as well as historical frauds. It consists in identifying unauthorized activity once the fraud prevention has failed. With reference to the MMT scenario proposed in the MASSIF project [11] we simulated the account takeover misuse case where a fraudster steals the mobile phone from the legitimate user and uses it to perform money transfer. More precisely, once the fraudster has stolen the mobile phone, he attempts to find the pin related to the mobile payment application. Usually the fraudster makes ten attempts with false PIN code to enter the system. Once the fraudster has gained access to the mobile payment application, he tries to do small purchases. To do that he moves from one merchant to another one in order to buy goods. The time interval between two transactions ranges from 3 to 20 seconds. The fraudster performs up to 30 transactions with an amount between 31 and 50 €. In order to detect the Account Takeover misuse case we propose a Fraud Detection System (FDS), which implements a number of rules to analyse the deviation of each incoming transaction from the normal profile of the user and assign an initial belief to it. The initial belief values are combined to obtain an overall belief by applying the Dempster–Shafer theory. The overall belief is then compared with two thresholds in order to understand if the user's behaviour is to be considered fraudulent or genuine. The proposed FDS comprises the following three major components: a Rule Based Filter, a Dempster-Shafer combiner and an Analyser. The flow of events in the FDS has been depicted in the block diagram in Fig. 1.

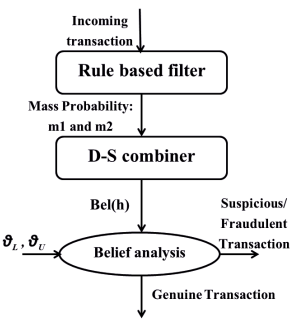


Fig. 2 Block diagram of the proposed FDS

4.1 The Rule Based Filter (RBF)

The RBF consists of rules which classify the transactions made by the user as fraudulent with a certain probability. It measures the extent to which the transaction's behavior deviates from the normal profile of the user. The rules used in our study are:

- Rule R1, authentication attempts: we analyse the time interval between the first and the last authentication attempt failed. If this time interval exceeds a given threshold (e.g. 15 seconds), then there is a high probability that the transaction is fraudulent.
- Rule R2, outlier detection: a user usually carries out similar types of transactions in terms of amount, which can be visualized as part of a cluster. Instead, a fraudster is likely to deviate from the customer's profile, so his transactions can be detected as exceptions to the cluster. This process is known as outlier detection. One of the most used algorithms used for detect cluster is the DBSCAN (*Density Based Spatial Clustering of Application with Noise*) [6]. Let $U^{'} = \{u_1 \dots u_n\}$ denote the clusters in a database *D* for a specific user of the MMT system U_k and $A = \{a_1 \dots a_n\}$ be the set of attributes used to generate the clusters. For any transaction the possible attributes are transaction amount, the date, the merchant involved in the transaction, etc. A transaction T^{c_k} is an outlier if it does not belong to any cluster in the set $U^{'}$. In this way we can understand if a transaction is fraudulent. The degree of outlierness allows to measure the extent of deviation of an incoming transaction. If the average distance of the amount *p* of an outlier transaction T^{c_k} from the set $U^{'}$ is v_{avg}, then its degree of outlierness is:

$$\begin{cases} d_{out} = \left(1 - \frac{\varepsilon}{v_{avg}}\right) & \mid N_\epsilon(p) \mid < MinPts \\ 0 & otherwise \end{cases}$$

  where *MinPts* is the minimum number of points required to form a cluster, while ε is the maximum radius of the cluster. As said earlier, to form a cluster we can use various attributes. In our study we used the amount of the transactions. Particularly, transactions with an amount between 31 and 50 €were considered as fraudulent.

An FDS is subjected to a large number of transactions, a high percentage of them being genuine. The RBF is an essential component since it separates out most of the easily recognizable genuine transactions from the rest.

4.2 The Dempster-Shafer Combiner (DSC)

The role of the DSC is to combine evidences from rules R1 and R2 and compute an overall belief value for each transaction. For the detection of fraud in the MMT system the Dempster-Shafer theory is more relevant as compared to other fusion

methods since it introduces a third alternative: "*unknown*". It provides a rule for computing the confidence measures of three states of knowledge: *{fraud, no fraud, suspicious}* based on data from new as well as old evidence. Furthermore, in DST, evidence can be associated with multiple possible events unlike traditional probability theory where evidence is associated with only one event. As a result, evidence can be more meaningful at a higher level of abstraction.

The part of DST that is of direct relevance is the Dempster's rule for combination [10]. In order to apply the Dempster-Shafer theory we need to define a frame of discernment U which is a set of mutually exclusive and exhaustive possibilities. With reference to the MMT fraud detection problem the frame of discernment is $U = \{\neg fraud, suspicious, fraud\}$. Hypothesis $F = \{fraud\}$ means that the transaction is fraudulent, hypothesis $N = \{\neg fraud\}$ is the hypothesis that the transaction is not fraudulent, and hypothesis $S = \{suspicious\}$ means that the transaction is suspicious. The mass probability assignments for the two rules R1 and R2 can now be given as follows:

- mass probability m_1: let *t* denote the time interval between the first and the last authentication attempt, we can consider the following assignments: if $t > 15$ seconds, then $[m_1(F) = 0.6, m_1(N) = 0, m_1(S) = 0.4]$. Instead, if $10 \leq t \leq 15$ seconds, then: $[m_1(F) = 0.4, m_1(N) = 0, m_1(S) = 0.6]$. Finally, if $t < 10$ seconds, then: $[m_1(F) = 0, m_1(N) = 0.6, m_1(S) = 0.4]$.
- mass probability m_2: for a transaction detected as an outlier we make the mass probability assignment using the degree of outlierness $d_{out} = 1 - \frac{\varepsilon}{v_{avg}}$ where ε is the credit limit that is the maximum amount of credit that a user can spend, while v_{avg} is the average distance of the amount of an outlier transaction from the set of the other transactions. Hence we consider the following assignment: $\left[m_2(F) = 1 - \frac{\varepsilon}{v_{avg}}, m_2(N) = 0, m_2(S) = 1 - \left(1 - \frac{\varepsilon}{v_{avg}}\right)\right]$.

As we can see in both cases the zero in the basic probability assignment for the hypothesis N does not imply impossibility. It means that neither of the rules R1 and R2 give any support to the belief that the set of transactions are genuine.

4.3 The Analyser

The two probability masses are combined using the Dempster-Shafer combiner to get the initial value of belief for the set of transactions made by the user. Particularly in our study we used the *Bel(F)*, i.e. the minimum probability that the event "*Fraud*" occurs. In our analysis we defined two thresholds: θ_L is the lower threshold, where $0 \leq \theta_L \leq 1$, and θ_U is the upper threshold, where $0 \leq \theta_U \leq 1$ and $\theta_L \leq \theta_U$.

If $Bel(F) < \theta_L$ the user behaviour is considered as genuine and is approved. On the other hand, if $Bel(F) > \theta_U$, then the user behaviour is declared to be fraudulent. In case $\theta_L \leq Bel(F) \leq \theta_U$, the user behaviour is labelled as suspicious.

The two thresholds and the other parameters can be chosen by observing the performance of the FDS over a large number of simulation trials.

5 Experimental Tests and Results

We demonstrated the effectiveness and performance of our FDS by conducting an extensive experimental campaign. Due to the unavailability of real data we used the simulator developed by the MASSIF project to generate synthetic transactions that represent the behaviour of genuine users as well as that of fraudsters [7]. We used standard metrics to evaluate the performance of the system under different test cases. True positives (TP) are the fraudulent users detected by the system and false positives (FP) are the genuine users with a normal behavior detected as fraudsters.

The effectiveness of the proposed system depends on θ_L and θ_U. If θ_U is set too high, then most of the frauds will go undetected, whereas if θ_U is set too low, then there will be a large number of false alarms. Similarly, high value of θ_L will let most of the frauds go through and low value of θ_L will lead to unnecessary investigation of a large number of genuine transactions. Hence, selection of the two thresholds has an associated tradeoff. We carried out our experiments to determine a good choice of these parameters.

In Fig.3 (left and right), we show how the mean values of TP and FP vary with each threshold value. Particularly, the values of TP strongly depend on the value of the upper threshold and this behaviour is especially noticeable for users who are victim of fraud. Instead, the values of FP depend on the value of the lower threshold and this behaviour is especially noticeable for users who are not a victim of fraud. It has to be noted that mean values of TP increase as θ_U increases. Good performance is attained with values of the upper threshold between 0.72 and 0.74. Instead, values of FP decrease as the θ_L increases, then good values for the lower threshold are under 0.35. The effectiveness of the FDS is also dependent on the two parameters, i.e. ε and *MinPts*. More precisely, the larger ε, the less is the number of clusters formed. In the limit, there will be only one large cluster. Also, the higher the value of MinPts, the less is the number of clusters formed. If it is set too high, there will be no clusters since the *MinPts* condition is not satisfied. However, if both the parameters are small, there can be a lot of clusters. If *MinPts* is set to 1, then each point in the database is

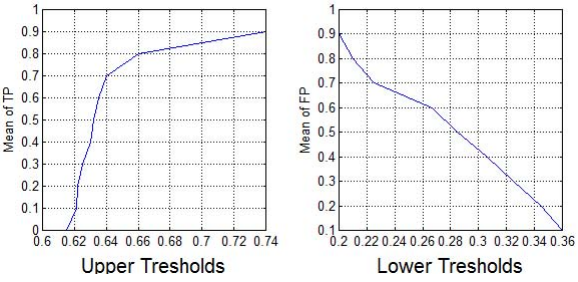


Fig. 3 Mean of True Positive (TP, left) and Mean of False Positive (FP, right) rates

treated as a separate cluster. In our study, after having studied the trend of the *Bel*, we decided to set $\varepsilon = 2\%$ of credit limit and $MinPts = 1$.

The experimental results show that the 95% of the users indicated by the simulator as victim of fraud is properly detected, while the 5% is detected as suspicious. Similarly, the 97% of the genuine users is properly detected, while the 3% is detected as suspicious.

6 Conclusions and Future Work

We have proposed a novel Fraud Detection System based on the integration of two approaches, i.e. the Dempster-Shafer theory and the rule-based filtering. Dempster's rule is applied in order to combine multiple evidences from the rule-based component for computation of belief about the transactions carried out by a user of the MMT system. This value of belief is compared with two thresholds in order to understand if the behaviour of the user is fraudulent, genuine or suspicious. Moreover the FDS has been designed as a modular architecture so that new rule-based filters can be added at a later stage using any other effective technique. The results of the simulation campaign show that the fraud detection system based on the Dempster-Shafer theory is able to detect frauds and suspicious behaviours of the MMT users. The system can be further improved by using a Bayesian approach for a more accurate assessment of the cases where the user is detected as suspicious. Finally, it would be interesting to compare the performance and accuracy of the Fraud Detection System based on the Dempster-Shafer theory with those of the FDS implemented by the MASSIF project and using the finite state machine technology. The latter approach has been already used in the scenario of an eHealth [4] infrastructure and of a dam infrastructure [2], [12].

Acknowledgements. The research leading to these results has received funding from the European Commission within the context of the Seventh Framework Programme (FP7/2007-2013) under Grant Agreement No. 313034 (Situation AWare Security Operation Center, SAWSOC Project) and under Grant Agreement No. 257644 (MAnagement of Security information and events in Service Infrastructures, MASSIF Project). It has been also partially supported by the TENACE PRIN Project (n. 20103P34XC) funded by the Italian Ministry of Education, University and Research.

References

1. Abidi, M.A., Gonzalez, R.C.: Data fusion in robotics and machine intelligence. Academic Press Professional, Inc. (1992)
2. Coppolino, L., D'Antonio, S., Formicola, V., Romano, L.: Enhancing siem technology to protect critical infrastructures. In: Hämmerli, B.M., Kalstad Svendsen, N., Lopez, J. (eds.) CRITIS 2012. LNCS, vol. 7722, pp. 10–21. Springer, Heidelberg (2013)
3. Arthur, P.: Dempster. A generalization of bayesian inference. Technical report, DTIC Document (1967)

4. Di Sarno, C., Formicola, V., Sicuranza, M., Paragliola, G.: Addressing security issues of electronic health record systems through enhanced siem technology. In: 2013 Eighth International Conference on vailability, Reliability and Security (ARES), pp. 646–653 (September 2013)
5. Durrant-Whyte, H.: Multi Sensor Data Fusion. Australian Centre for Field Robotics (2001)
6. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, vol. 96, pp. 226–231 (1996)
7. Gaber, C., Hemery, B., Achemlal, M., Pasquet, M., Urien, P.: Synthetic logs generator for fraud detection in mobile transfer services. In: 2013 International Conference on Collaboration Technologies and Systems (CTS), pp. 174–179. IEEE (2013)
8. Gros, X.: NDT data fusion. Elsevier (1996)
9. Jannin, P., Grova, C., Gibaud, B.: Medical applications of NDT data fusion. Springer, Heidelberg (2001)
10. Panigrahi, S., Kundu, A., Sural, S., Majumdar, A.K.: Credit card fraud detection: A fusion approach using dempster–shafer theory and bayesian learning. Information Fusion 10(4), 354–363 (2009)
11. Rieke, R., Coppolino, L., Hutchison, A., Prieto, E., Gaber, C.: Security and Reliability Requirements for Advanced Security Event Management. In: Kotenko, I., Skormin, V. (eds.) MMM-ACNS 2012. LNCS, vol. 7531, pp. 171–180. Springer, Heidelberg (2012)
12. Romano, L., D'Antonio, S., Formicola, V., Coppolino, L.: Protecting the WSN zones of a critical infrastructure via enhanced SIEM technology. In: Ortmeier, F., Daniel, P. (eds.) SAFECOMP Workshops 2012. LNCS, vol. 7613, pp. 222–234. Springer, Heidelberg (2012)
13. Shen, S.: Market insight: The outlook on mobile payment (2010)
14. Smith, D., Singh, S.: Approaches to multisensor data fusion in target tracking: A survey. IEEE Transactions on Knowledge and Data Engineering 18(12), 1696–1710 (2006)
15. Srivastava, R.P.: The dempster-shafer theory: An introduction and fraud risk assessment illustration (2011)
16. Zhang, J.: Multi-source remote sensing data fusion: status and trends. International Journal of Image and Data Fusion 1(1), 5–24 (2010)

An Hybrid Architecture to Enhance Attacks Detection on IT infrastructure*

Mario Sicuranza, Giovanni Paragliola,
Cesario Di Sarno, and Alessia Garofalo

Abstract. Nowadays, IT systems are widely used to support the services offered from any infrastructure. This allows the improvement of business processes but on the other hand it exposes the infrastructure to cyber-attacks. Misuse and anomaly detection are two widely adopted approaches to discover known and unknown cyber-attacks. In this paper we provide an overview of the techniques currently adopted for misuse and anomaly detection and we discuss a conceptual architecture that exploits the advantages of both misuse and anomaly detection to improve cyber-security. Also we provide a conceptual description of an expert system that solves conflicts due to detection mismatches between misuse and anomaly detection techniques.

Keywords: Misuse detection, Anomaly detection, Expert System.

1 Introduction

IT systems are increasingly being adopted in several domains, even in those where highly critical and/or highly sensitive information is managed such as in Critical Infrastructures [1] and in healthcare [2]. In these domains, a correct and complete detection of cyber attacks has to be ensured. An unnecessarily high amount of security alerts can cause a human/automatic system to process lots of irrelevant data and detect the relevant ones too late (or even discard them). Another issue of IT systems is the heterogeneity of cyber threat sources. In fact, data relevant to the specific

Mario Sicuranza, Giovanni Paragliola
Institute of High Performance Computing and Networking, National Research Council, ICAR-CNR, Naples, Italy
e-mail: {mario.sicuranza,giovanni.paragliola}@na.icar.cnr.it

Cesario Di Sarno · Alessia Garofalo
University of Naples "Parthenope", Department of Engineering, Naples, Italy
e-mail: {cesario.disarno,alessia.garofalo}@uniparthenope.it

* This work has been partly supported by the project ÒSmart-Health 2.0Ó (PON04a2_C/20).

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII,*
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_45

domain considered (business domain data) are available at the application level, so this level has to be protected from cyber attackers by trying to modify data from applications able to access those data. At the same time, attackers often attempt to access data by compromising the system in different ways, e.g. by attacking the routing level [1]. In this work, a new architecture has been designed whose purpose is to ensure cyber security by combining knowledge of the business domain and of the cyber threats that can occur. In the architecture, both anomaly and misuse based detection are used to take advantage of the detection capabilities of both 0-day attacks and well-known attacks. The architecture also provides a mitigation of the disadvantages of anomaly and misuse detection (e.g. high false positives and the incapability of detecting previously described attacks e.g. through signatures). This is obtained through an Expert System, which has a detailed knowledge of the specific business domain considered and so can improve detection of cyber attacks. The purpose of this work is to present a conceptual architecture; in future works this will be implemented and tested in a real scenario. The remainder of the paper is detailed as follows. In Section 2 the commonly used detection techniques for cyber attacks are detailed; in Section 3 the conceptual architecture is described. Finally, in Section 6 concluding remarks and future works are discussed.

2 Related Work

This section provides a brief overview about the state of the art of analysis techniques adopted in Intrusion Detection Systems (IDSs). The literature provides a lot of papers that describe monitoring systems based on intrusion detection techniques such as [13] and [14]. A classic classification defines two basic approaches: *Misuse strategies* and *Anomaly strategies*. In the case when an IDS looks for events or sets of events that match a predefined signature of a known attack, we refer to Misuse Detection strategy; instead we refer to Anomaly Detection strategy when the IDS identifies intrusions as unusual behavior that differs from the normal behavior of the monitored system. **Misuse Detection Techniques.** In *signature-based* IDSs, events are monitored and matched against a database of known attack signatures to detect intrusions. In [2], the authors adopt a signature-based approach to discover business logic anomalies and protect the identities of involved parties of a Electronic Health Record (EHR) system. *Rule-based* systems use a set of âĂIJif-thenâĂİ implication rules to characterize cyber attacks. In this kind of IDSs, events are monitored and then coded in facts and/or rules that are later used by an inference engine to draw conclusions and claim if an attack is occurring or not. An example of such kind of IDSs is proposed in [3], where the authors propose a novel intrusion detection framework for securing wireless sensor networks from routing attacks. IDSs based on *state transition* techniques define a finite state machine that models the system's evolution over the input; in such model, each state corresponds to different IDS states, and each transition describes certain events that cause IDS states to change. In [8], the authors model penetrations as a series of state changes that lead from an initial secure state to a target compromised state. **Anomaly Detection Techniques.** *Classification*

based techniques define a model from a dataset of labeled instances, so called *training*; the learnt model is then used to classify test instances in one or more classes. The simplest classification model entails a binary classification as {*normal*, *abnormal*}. An example of such kind of approach was brought by [4], where the authors propose a novel approach to detect anomalous records in categorical datasets based on Bayesian Networks. *Statistical based techniques* produce a statistical model from given data; such model embodies the normal behavior of the system monitored. After the model is built, incoming instances are tested in order to check if the current instance belongs to the model or not. When an instance does not match the model, it is marked as an anomaly. An example of this approach is provided by EMERALD [5], which defines a statistical profile-based anomaly detection module that tracks activities through one of four types of statistical variables: categorical, continuous, traffic intensity, and event distribution. In *rule-based* approaches, a knowledge base is defined which describes the normal system behavior, so the anomaly evaluation is performed by comparing this predefined normal behavior with the system current activities. In [6], the authors enhance this approach by introducing a threshold to prune out rules with low support. *Profile based techniques* are grounded in the definition of a profile or description of the system that we need to monitor. The profile describes the activities of the system by means of attributes. A typical example is fraud detection; [7] proposes a credit card fraud detection model using outlier detection based on the distance sum according to the infrequency and unconventionality of fraud in credit card transaction data. Research in anomaly detection also focuses on the classification of intrusions by applying various standard *data mining* algorithms to data collected from the monitored system. In [9], the authors present an approach based on data mining that supports signature discovery in a network-based IDS. In [11] authors investigate the use of *machine learning* and *soft computing* methodologies to perform intrusion detection.

3 Proposed Approach

In Figure 1 we show the conceptual architecture proposed to ensure cyber-security of a generic IT infrastructure monitored. A typical IT infrastructure is composed of: software applications; an IT layer to support software applications (e.g. web server, database and so on); network devices that allow communication with other hosts/devices. The components of the IT infrastructure output several *logs*, which contain information related to the corresponding component, e.g. web server status, system calls performed, firewall status and so on. Logs can be useful to perform security analysis and to avoid/prevent security breaches. Logs generated from any source within the monitored infrastructure are gathered by different software *agents*. Software agents perform a preliminar security analysis and send their output (*events*) to *Misuse* and *Anomaly Detection* modules. Both the modules are equipped with a corresponding Knowledge Base, which is used to perform a more *complex* and *complete* analysis compared to each single agent; this is because information related to events from different locations are available to the modules. Both modules independently analyze events generated by agents to discover suspicious activities. If the outputs

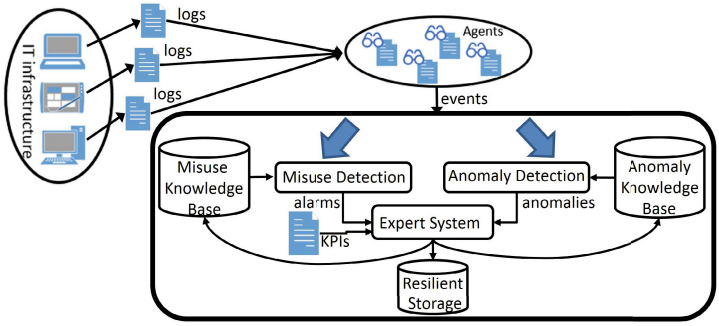


Fig. 1 Architecture proposed to protect an infrastructure from cyber-attacks

provided by both modules disagree then the *Expert System* is invoked. The Expert System (or human expert) resolves the conflict, decides if an attack is occurring or not and updates the knowledge base of the modules (if necessary). Finally, if an attack occurs, Expert System stores alarms/anomalies raised in secure way through *Resilient Storage*. The Resilient Storage is a facility designed to be tolerant to intrusions and faults [10]. The components of the architecture proposed are detailed in the following.

3.1 Agents

Agents are software modules deployed within the monitored IT infrastructure. In particular, agents must be designed to gather and to process a huge amount of logs and eventually to handle even traffic peaks which can cause overload on a specific agent. These requirements can be satisfied through a federated agents model which dynamically optimizes allocation of resources and available agents. Each data source within the monitored infrastructure typically generates logs in a specific format chosen by the corresponding producer. This makes the logs analysis much harder when heterogeneous sources are provided in the system. To overcome this limitation, each agent performs a *normalization* process i.e. all logs gathered are translated in *events* with a common format and a specific semantic. Finally, each agent uses its own local knowledge provided by such events to perform fine-grained security analysis.

3.2 Misuse and Anomaly Detection

Events generated by agents are sent to *Misuse* and *Anomaly Detection* modules. Misuse Detection module allows to detect well-known attacks, e.g. attacks widely known that can be described through signatures.The advantage of such kind of approach is that well-known attacks are accurately identified. On the other hand, an attack cannot be detected successfully if a description of the attack signature is not available. To overcome this limitation we propose to support Misuse Detection module with an Anomaly Detection module. Anomaly Detection is a complementary approach

Table 1 The behavior of the Expert System with reference to outputs of Anomaly and(or) Misuse Detection modules

Misuse Detection	Anomaly Detection	Effect
0	0	None
1	0	Update the knowledge base of the anomaly or misuse module
0	1	Update the knowledge base of the anomaly or misuse module
1	1	Stores alarms/anomalies in Resilient Storage

to misuse that allows to discover new or unknown attacks. Anomaly Detection is based on the concept of baseline. In particular a baseline describes the normal behavior of the phenomenon monitored. A known disadvantage of anomaly detection is that it can generate many false positives. To overcome this limitation, the Anomaly Detection module sends the same events to different sub-modules that use different techniques of data analysis to discover anomalies. Finally a voting system allows to select the best output provided by sub-modules. Both profiles generated and well-known attack patterns are described and stored respectively in the anomaly and misuse knowledge bases.

3.3 Expert System

The purpose of the *Expert System* is to take advantage of both Misuse and Anomaly Detection modules. The Expert System makes use of high-level policies defined by an expert of the specific domain considered. Key Performance Indicators (KPIs) are extracted from policies; they represent the attributes that must be monitored to enforce policies. The Expert System has three information sources: i) alarms raised by Misuse Detection module; ii) anomalies raised by Anomaly Detection module; iii) KPIs extracted from policies. The Expert System processes alarms/anomalies coming from the Misuse Detection and the Anomaly Detection modules as described in Table 1. If both Misuse and Anomaly Detection modules do not raise any alarm/anomaly (first row in the table), the Expert System does not perform any action. If the Misuse Detection module raises an alarm related to a detected attack and the Anomaly Detection module does not detect any anomaly (second row in the table) or if the Misuse Detection module does not raise any alarm and the Anomaly Detection module raises an anomaly (third row in the table), then a conflict is generated and the Expert System is invoked. In this case the Expert System has to resolve the conflict in order to decide if an attack in occurring or not. Thus, it evaluates the impact of alarm/anomaly identified through KPI previously established. As consequence of its evaluation, the Expert System will update the knowledge base of the Anomaly or Misuse Detection module. Furthermore the Expert System stores the alarm related to the attack detected within the Resilient Storage. The last case of Table 1, i.e. when

both modules generate an alarm/anomaly, is referred to the case when the attack is recognized and so this alarm/anomaly is stored in Resilient Storage.

4 Conclusion

The aim of this paper is to present a solution to combine existing solutions for anomaly and misuses detection for the detecting of attacks on IT system. In this paper we have defined a general architecture to use the advantages of both approaches and reduce the limits of such kind approaches. The proposed general architecture will be applied to a specific domain (e.g the Healthcare domain) in order to define, implement and validate the components of the whole architecture (the agents, the detection module and the expert system). As future work we will build a prototype of the solution in a real healthcare domain(e.g Electtronich Health Record System [12]).

References

1. Coppolino, L., D'Antonio, S., Garofalo, A., Romano, L.: Applying Data Mining Techniques to Intrusion Detection in Wireless Sensor Networks. In: P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC) (2013)
2. Di Sarno, C., Formicola, V., Sicuranza, M., Paragliola, G.: Addressing Security Issues of Electronic Health Record Systems through Enhanced SIEM Technology. In: Availability, Reliability and Security, ARES (2013)
3. Eswari, T., Vanitha, V.: A novel rule based intrusion detection framework for Wireless Sensor Networks. In: ICICES (2013)
4. Das, K., Schneider, J.: Detecting anomalous records in categorical datasets. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2007 (2007)
5. Neumann, P., Porras, P.: Experience with Emerald to Date. In: Proceedings of the First Workshop on Intrusion Detection and Network Monitoring, Santa Clara (1999)
6. Tandon, G., Chan, P.K.: Weighting versus pruning in rule validation for detecting network and host anomalies. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2007)
7. Yu, W.-F., Wang, N.: Research on Credit Card Fraud Detection Model Based on Distance Sum. In: Artificial Intelligence, JCAI 2009 (2009)
8. Ilgun, K., Kemmerer, R.A., Porras, P.A.: State transition analysis: a rule-based intrusion detection approach. IEEE Transactions Software Engineering
9. Han, H., Lu, X.-L., Ren, L.-Y.: "Using data mining to discover signatures in network-based intrusion detection. Machine Learning and Cybernetics (2002)
10. Afzaal, M., Di Sarno, C., Coppolino, L., D'Antonio, S., Romano, L.: A Resilient Architecture for Forensic Storage of Events in Critical Infrastructures. In: Proceedings of the 2012 IEEE 14th International Symposium on High-Assurance Systems Engineering (2012)
11. Camastra, F., Ciaramella, A., Staiano, A.: Machine learning and soft computing for ICT security: an overview of current trends. J. Ambient Intelligence and Humanized Computing 4(2), 235–247 (2013)
12. Sicuranza, M., Ciampi, M., De Pietro, G., Esposito, C.: Secure healthcare data sharing among federated health information systems. Int. J. Crit. Comput.-Based Syst. 4(4), 349–373 (2013)
13. Ficco, M.: Security event correlation approach for cloud computing. International Journal of High Performance Computing and Networking (IJHPCN) 7(3) (2013)
14. Ficco, M., Coppolino, L., Romano, L.: A Weight-Based Symptom Correlation Approach to SQL Injection Attacks. In: Fourth Latin-American Symposium on Dependable Computing, LADC 2009, September 1-4 (2009)

Author Index